NASA Contractor Report 170395

NASA-CR-170395
19830010473

Flight Test Trajectory Control Analysis

Robert Walker and Naren Gupta

Contract NAS4—2906
February 1983

National Aeronautics and
Space Administration

All Blank Pages in this Document

are Intentionally Left Blank

NASA Contractor Report 170395

43 1 1 RN/NASA-CR-170395

DISPLAY 43/2/1

83N18744*# ISSUE 9 PAGE 1312 CATEGORY 8 RPT#: NASA-CR-170395 NAS 1.26:170395 151-16 CNT#: NAS4-2906 83/02/00 138 PAGES UNCLASSIFIED DOCUMENT

UTIL: Flight test trajectory control analysis TLSP: Final Report

AUTH: A/WALKER, R.; B/GUPTA, N.

CORP: Integrated Systems, Inc., Palo Alto, Calif. AVAIL.NTIS SAP: HC A07/MF A01

MAJS: /*AIRCRAFT MANEUVERS/*FLIGHT TESTS/*LINEAR SYSTEMS/*PILOT PERFORMANCE/* TRAJECTORY CONTROL

MINS: / APPROXIMATION/ COMPUTER PROGRAMS/ COMPUTERIZED SIMULATION/ F-15 AIRCRAFT / TRANSFER FUNCTIONS/ WORKLOADS (PSYCHOPHYSIOLOGY)

ABA: Author

ABS: Recent extensions to optimal control theory applied to meaningful linear models with sufficiently flexible software tools provide powerful techniques for designing flight test trajectory controllers (FTTCs). This report describes the principal steps for systematic development of flight trajectory controllers, which can be summarized as planning, modeling

ctory

s

National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523

N83-18744#

FOREWORD

The report discusses the development of a systematic methodology to aid pilots in accurately following aircraft flight test trajectories. Modern control methods are used to compute time histories for displays which both reduce pilot workload and make test trajectories consistently accurate.

The work has been performed under NASA Contract NAS4-2906. Technical discussion and direction of NASA technical monitor Mr. E. L. Duke are gratefully acknowledged.

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

SECTION 1

INTRODUCTION

It is often necessary to follow desired time histories of several variables during an aircraft flight test. The pilots often find it difficult to follow such trajectories accurately, leading to increased test time and pilot workload. This report shows that modern control techniques can be applied to significantly improve the accuracy with which complex trajectories can be flown. In addition, the pilot inputs are concentrated below 2 rad s^{-1} reducing pilot workload and fatigue.

The study involves computing three display variables on which pilot pitch, roll and throttle commands can be based. The pilot perceives a three axis uncoupled control problem because all couplings are handled by the flight trajectory controller.

The emphasis has been placed on the development of the techniques. An F-15 nonlinear simulation has been used for demonstration. The techniques are general and should be applicable to various combinations of trajectories, aircraft and displays.

1.1 SUMMARY OF APPROACH

The principal steps for systematic development of flight trajectory controllers are

1. Development of the flight test plan,
2. Control design model specification,
3. Controller design, and
4. Controller validation by analytical simulation and pilot-in-the-loop-simulation.

The flight test plan prescribes the aircraft with all subsystems configured for the given test, the trajectory or classes of trajectories to be flown, and the pilot's tasks during the specified trajectory maneuvers. Figure 1-1 depicts the design procedure after the flight maneuver and system models have been specified.

1.2 RESULTS

The major result from this study is a set of tools suitable for developing flight test trajectory controllers. Attention has been given to keep the techniques as general as possible. The techniques should have a general application to a wide class of problems where quantities displayed to a pilot should be appropriately computed to improve pilot effectiveness and to reduce workload and fatigue.

The development of the control laws is straightforward using an advanced control design program once the model and the performance index have been specified. Thus, major part of the attention must be given to these two tasks. Special linearization tools were developed for the F-15 simulation to develop a suitable model for control. Integration of pilot model is a difficult task because the pilot model itself depends upon the trajectory controller. This was achieved by prespecifying a suitable representation for the aircraft and controller dynamics and basing a pilot model on this dynamics. It is necessary to roll-off the control law at high frequency, and frequency shaping methods are essential to achieve necessary gain margin at high frequency.

Pilot-in-the-loop and flight test validation of controllers are needed. It may be desirable to integrate pilot inputs with autopilots to achieve even better overall performance.

1.3 REPORT ORGANIZATION

This report is organized as follows: Sections 2 through 4 describe systematic methods for designing flight test trajectory

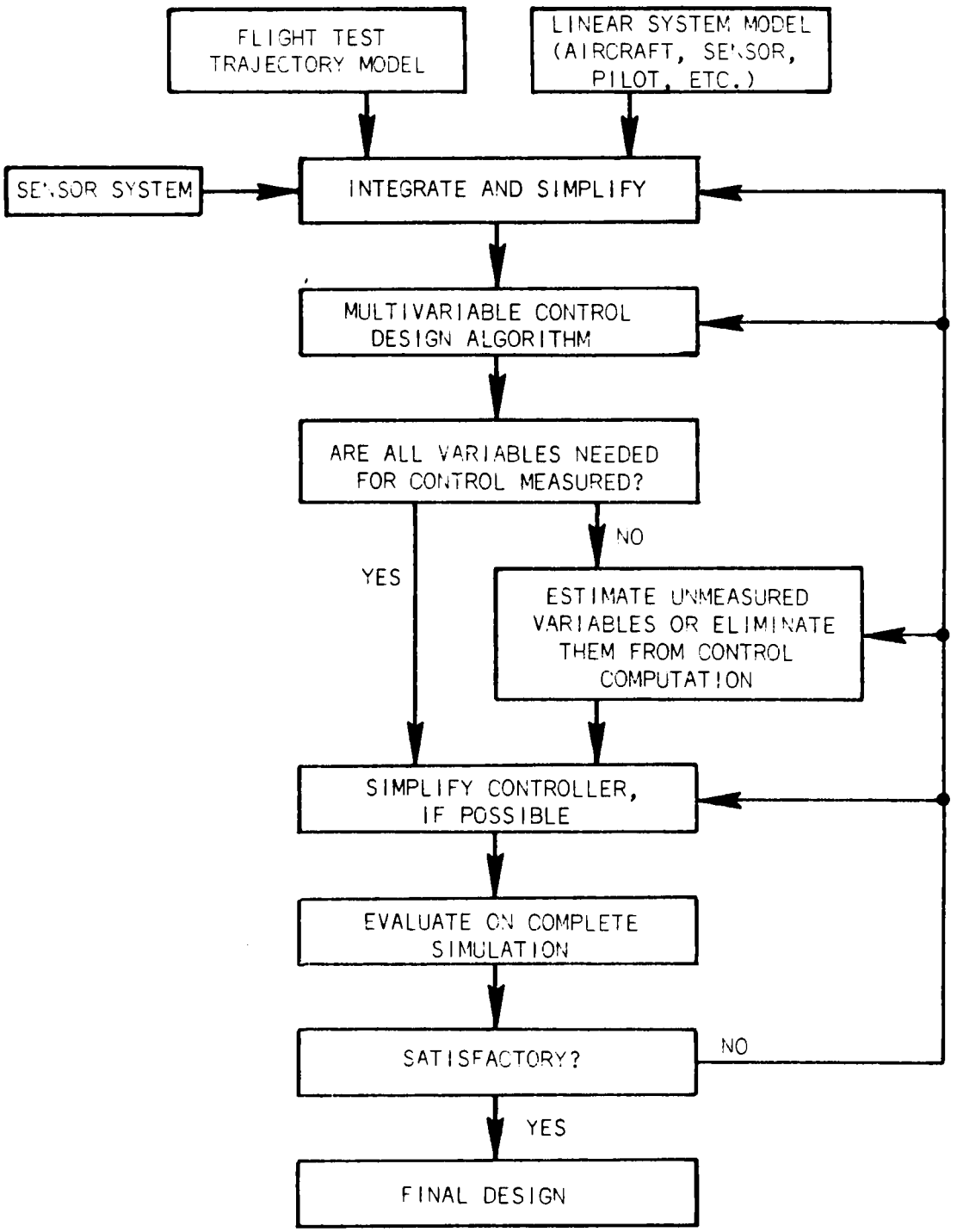

Figure 1-1 Flight Test Trajectory Control Design Procedure

controllers, illustrated on the F-15 aircraft. Section 2 categorizes how the equations are developed for different types of maneuvers. Section 3 develops linear models for each major subsystem, and Section 4 describes the control design methods.

Section 5 describes the control law synthesis for the zoom and pushover maneuver, and linear and nonlinear evaluation of the control law. Section 6 documents this control algorithm for use in a batch simulation, manned simulator or flight test.

General conclusions are given in Section 7.

SECTION 2

TRAJECTORIES FOR FLIGHT TEST MANEUVERS

A large class of trajectories can benefit from flight test trajectory controller design. The model for the control law and the development of the control law will depend strongly on the particular trajectory to be flown.

Aircraft flight test trajectories could be based on inertial reference (e.g., level-turn or 3-D guidance) or reference with respect to another vehicle or vehicles (e.g. in air-to-air combat). To place the flight test trajectory control design problem in a proper framework, both of these cases will be discussed.

2.1 SINGLE VEHICLE FLIGHT TEST TRAJECTORY

We divide single vehicle flight paths into those which require continuous control along the trajectory and those that specify a final flight condition. In either case the flight test trajectory could be specified in terms of one of the following:

1. constraints on position components,
2. constraints on velocity components and altitude,
3. constraints on combinations of load, speed and altitude.

Combinations of these constraints could also be considered.

Constraints on Position Components

Examples of trajectories which involve position constraints along the flight path are

1. 4-D guidance (x(t), y(t), h(t) are given functions of of time).

2. 3-D guidance (x, y, h are related to each other, e.g., fly along a hypothetical wire in space). Examples of 3-D guidance are approach to landing, terminal area flight paths and threat evasion for reconnaissance aircraft and bombers. This also includes straight and level flight and flights along predetermined paths.

Examples of trajectories which specify position constraints at final trajectory point are:

1. 4-D specification (arrive at a certain point, at a certain time, e.g., touchdown on the runway at a specified point).

2. 3-D specification (fly-to-VOR, terrain following).

Note that each of these trajectories requires position measurement. The 4-D guidance trajectory indirectly specifies velocity and acceleration components. Thus specification of position components is the most comprehensive constraint on the trajectory. Such a rigid constraint may be unnecessary for most test maneuvers.

Constraints on Velocity Components and Altitude

While the horizontal position components do not, in general, affect aerodynamic variables, the altitude determines density and by itself affects dynamic pressure. Thus it must always be considered as a possible variable to be constrained. In fact, the altitude and dynamic pressure are so important that a majority of flight test trajectories will define the altitude profile (this includes maintaining constant altitude).

Examples of this class of trajectories are:

1. u(t), v(t), w(t) and h(t) [in other words, Mach number, dynamic pressure, $\alpha(t)$ and $\beta(t)$. $\beta(t)$ may be zero.

2. Mach number, angle-of-attack and dynamic pressure (as in shuttle tile tests).

Various other combinations of velocity components and altitudes could also be specified.

Mach number, angle-of-attack and altitude constraints could also be desired at one point on the trajectory. Zoom and pushover is a trajectory where angle-of-attack, Mach number and altitude are specified at one point on the trajectory. Note that since sideslip angle is assumed zero, zoom and pushover requires a certain set of u, v and w at a certain altitude.

Constraints on Combinations of Load, Speed and Altitude

The trajectory specifications could involve components of loads along the three axes, velocity components and altitude. The typical load specification will consist of desired vertical acceleration. The desired value of the lateral acceleration is usually zero. The total speed is specified in lieu of the fore-and-aft acceleration.

Many combinations of load, speed and altitude specifications are possible. Some examples are as follows:

1. A constant load, constant Mach number level turn,
2. A constant Mach number, constant altitude spiral turn.

2.2 MULTIPLE VEHICLE FLIGHT TEST TRAJECTORIES

Often the desired flight trajectory for an aircraft depends upon the position and flight test trajectory of other vehicles. Typical examples are collision avoidance or air combat with air-to-air missiles. The specification is typically based on the position of a target aircraft with respect to the aircraft whose trajectory is being controlled.

SECTION 3

SYSTEM MODEL FOR TRAJECTORY CONTROLLER DESIGN

The overall system model for the flight test trajectory control analysis has the following components:

1. aerodynamics and kinematics,
2. hydro-mechanical control system (MCS) and command augmentation system (CAS) including actuator models,
3. engines,
4. sensors, and
5. pilot.

The flowchart of the complete model is shown in Figure 1.

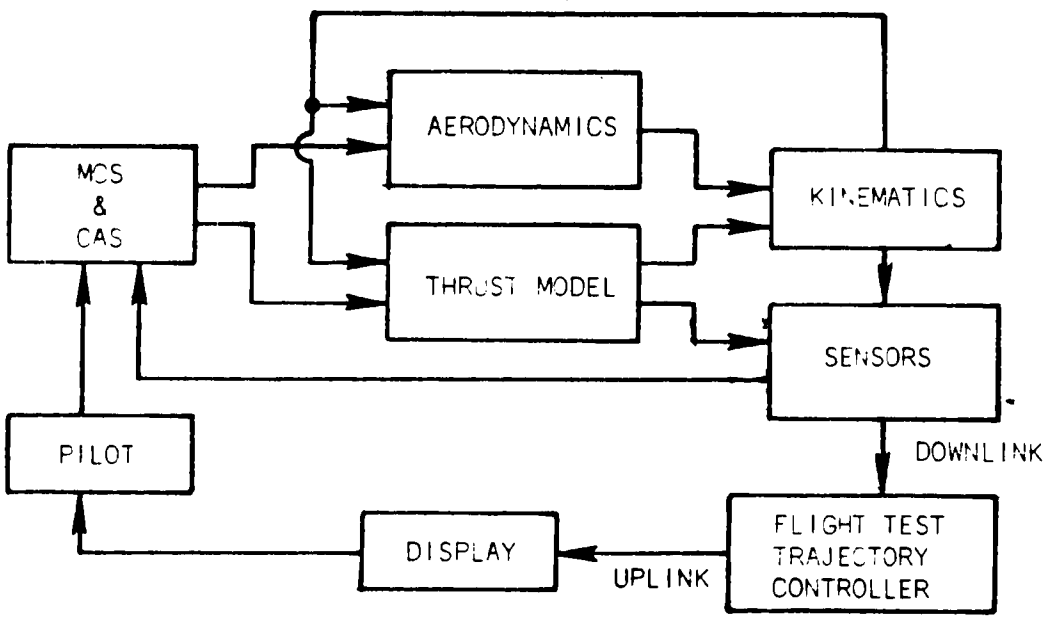

Figure 3-1 Schematic of System Model for Flight Test Trajectory Control Analysis

3.1 AERODYNAMICS AND KINEMATICS

The structure of mathematical equations governing the motion of a rigid aircraft may be derived from kinematics. Aerodynamics will be linearized, as necessary. A different approach must be used to smooth out any discontinuities in slope.

Consider an aircraft state vector

$$\underline{X} = \begin{bmatrix} \underline{x} \\ \underline{v} \\ \underline{\dot{\Phi}} \\ \underline{\Theta} \end{bmatrix} , \quad \text{with} \tag{3.1}$$

$$x = \begin{bmatrix} x \\ y \\ z \end{bmatrix} , \quad \underline{v} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} , \quad \Phi = \begin{bmatrix} p \\ q \\ r \end{bmatrix} , \quad \text{and}$$

$$\Theta = \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} . \tag{3.2}$$

The equations of motion can be written as

$$\dot{\underline{X}} = \underline{f}(\underline{X},\underline{u}) \tag{3.3}$$

A detailed nonlinear aerodynamic model of the F-15 from the NASA Dryden Flight Research Center Simulation SIMII was linearized by

a) trimming the aircraft at the desired flight condition,

b) perturbing the six aerodynamic force and moment equations with respect to all dynamic variables of interest, and

c) projecting these stability derivatives onto both body and stability axes (see Tables 5-1 and 5-2 in Subsection 5.2).

3.1.1 Longitudinal Aerodynamic Model

The simplified linear longitudinal equations developed further in Section 5.2 for the zoom and pushover maneuver are:

$$\delta\dot{V} = \left[- g\left(\frac{T_o}{W}\right)\sin \alpha_o - X_\alpha + g \right] \delta\alpha - g\delta\theta + \cos \alpha_o\left(\frac{\delta T}{m}\right) + X_\delta \delta_{SB}$$

$$\delta\dot{\alpha} = \frac{1}{V_o}\left\{ \left[g\left(\frac{T_o}{W}\right)\cos \alpha_o + Z_\alpha \right] \delta\alpha + q + \sin \alpha_o\left(\frac{\delta T}{m}\right) + Z_\delta \delta_{SB} \right\}$$

$$\dot{q} = \left\{ L_\alpha + L_{\dot{\alpha}}\left[g\left(\frac{T_o}{W}\right)\cos \alpha_o + Z_\alpha \right] \right\} \delta\alpha + (L_{\dot{\alpha}} + L_q)\, q + L_{\dot{\alpha}} \sin \alpha_o\left(\frac{\delta T}{m}\right) + (L_\delta + L_{\dot{\alpha}} Z_\delta)\, \delta_{SB}$$

$$\delta\dot{\theta} = q$$

$$\delta\dot{h} + V_o(\delta\theta - \delta\alpha) \ ,$$

where

δV - speed

$\delta\alpha$ - angle-of-attack

q - pitch rate

$\delta\theta$ - pitch attitude

δh - altitude

δ_{SB} - stabilator deflection

$\delta T/m$ - thrust acceleration

are perturbation states about the trimmed condition in stability axes.

3.1.2 Coupled Lateral/Longitudinal Equations

For high bank level turns trimmed at a constant turn rate, $\dot{\psi}_o$,

$$p_o = - \dot{\psi}_o \sin \theta_o$$

$$q_o = \dot{\psi}_o \cos \theta_o \sin \phi_o$$

$$r_o = \dot{\psi}_o \cos\theta_o \cos\phi_o \ .$$

The coupled linear equations in body axes are

$$\begin{bmatrix} \dot{v} \\ \delta\dot{r} \\ \delta\dot{p} \\ \delta\dot{\phi} \\ \delta\dot{w} \\ \delta\dot{q} \\ \delta\dot{\theta} \\ \delta\dot{h} \end{bmatrix} = \begin{bmatrix} Y_v & -U_o & w_o & g\cos\theta_o\cos\phi_o & 0 & 0 & 0 & 0 \\ N_v & N_r & N_p+p_z q_o & 0 & 0 & p_z p_o & 0 & 0 \\ L_v & L_r+r_x q & L_p & 0 & 0 & r_x p_o & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & p_o\theta_o\sin\phi_o & 0 & 0 \\ -p_o & 0 & 0 & -g\cos\theta_o\sin\phi_o & Z_w & U_o & 0 & 0 \\ M_v & q_y p_o & q_y r_o & 0 & M_w & M_q & 0 & 0 \\ 0 & -\sin\phi_o & 0 & -(r_o\cos\phi_o+q_o\sin\phi_o) & 0 & \cos\phi_o & 0 & 0 \\ -\sin\phi_o & 0 & 0 & w_o\cos\theta_o\sin\phi_o & -\cos\phi_o\cos\theta_o & 0 & U_o & 0 \end{bmatrix}$$

where $r_x = (I_y-I_z)/I_x$, $q_y = (I_z-I_x)/I_y$, $p_z = (I_x-I_y)/I_z$.

3.2 MCS AND CAS MODELS

The mechanical control system (MCS) and control augmentation system (CAS) use pilot commands and instrumentation system measurements as inputs. The outputs of this system are the aerodynamic control surface deflections and the thrust command.

Only those dynamics in the pilot's range of attention and response need to be modeled (.1 to 20 radians per second). Symmetric simple nonlinearities such as position and rate limits have no memory and should be avoided in trajectory controllers.

An overview schematic of the longitudinal MCS/CAS system is given in Figure 3-2. The mach servo loop is fixed at a constant gain obtained from SIMII at the trimmed condition. Describing function analysis indicated that the nonlinear control elements could be realized at the zoom and pushover trimmed condition as shown in Figure 3-3 (see Section 5.2 for the linear equations).

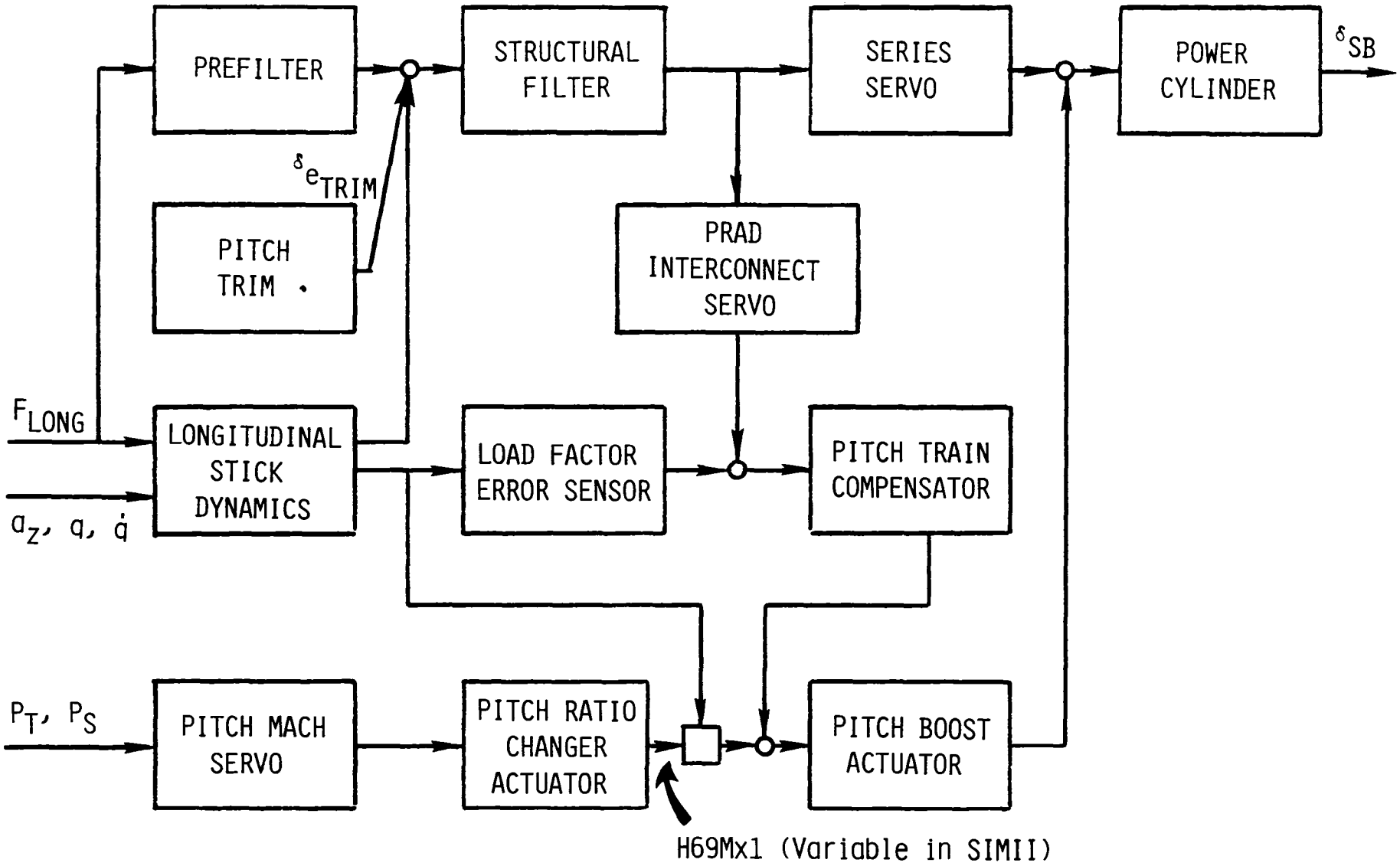

Figure 3-2 F-15 Longitudinal MCS/CAS

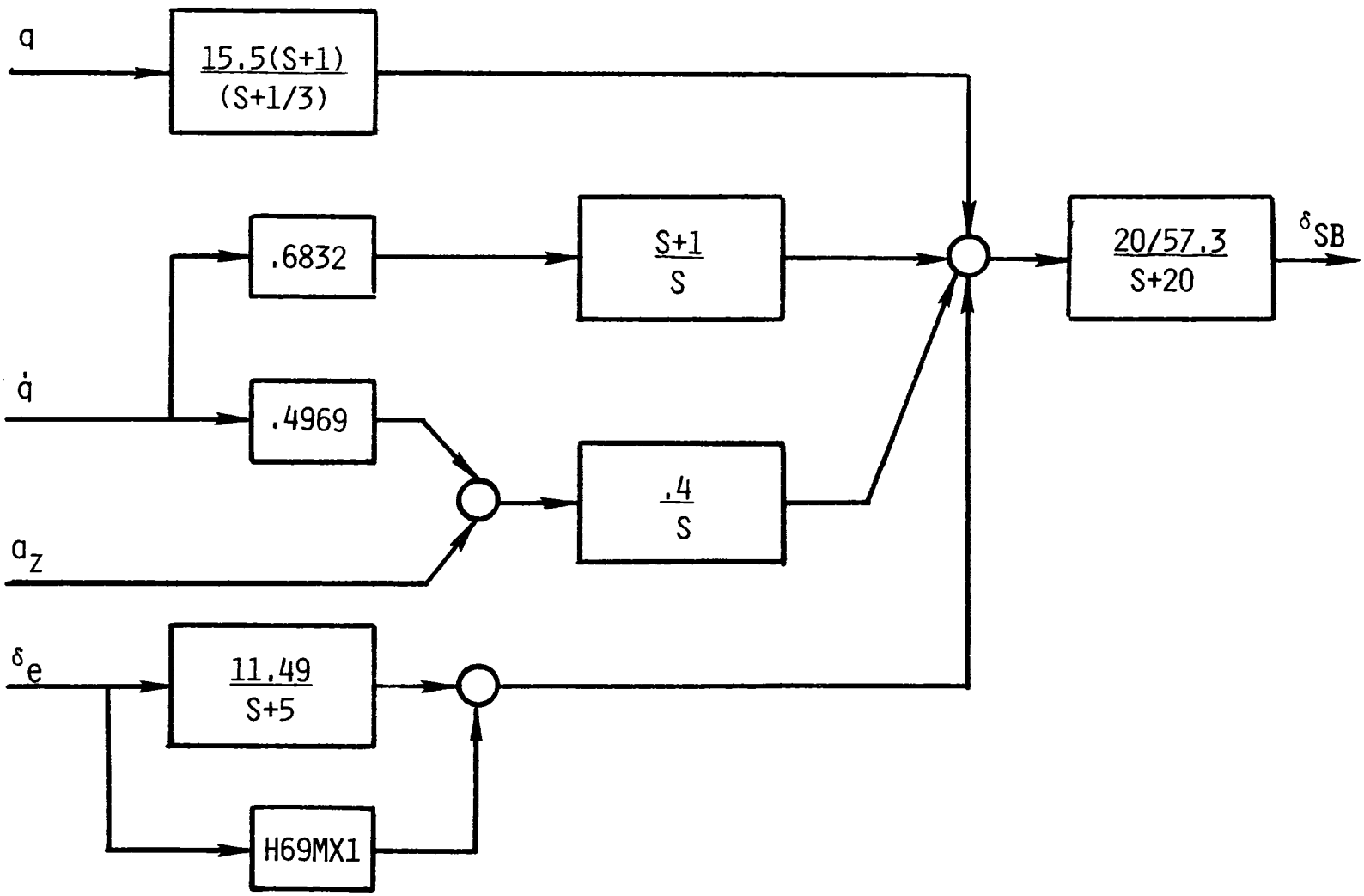

Figure 3-3 F-15 MCS/CAS Longitudinal Linear Model (Roll, Pitch, Yaw CAS on)

3.3 ENGINE MODEL

Simple as well as highly sophisticated models have been developed for engines such as F100-PW-100. A simple first-order lag model should suffice for the purpose of the study. The engine thrust T is related to the commanded engine thrust T_C as

$$\dot{T} = -\frac{1}{\tau} T + \frac{1}{\tau} T_C \ , \tag{3.4}$$

where τ is the time constant in the first-order lag model.

3.4 SENSOR MODEL

Typical instruments on board a test aircraft include a three-axis accelerometer, a three-axis rate gyro, angle gyros, angle-of-attack and sideslip angle vanes, altitude, rate-of-climb and total speed.

We could use a model with first-order lag and random noise. Thus, the measured quantity y_m will be related to the true y as follows

$$\tau \dot{y}_m = y - y_m + n \tag{3.5}$$

where τ is the time constant and n is the random noise. All sensors give sampled measurements (with an assumed rate of 10 Hz).

3.5 PILOT TASK DEFINITION

The trajectory models described in the previous section specify a portion of the controller task; however, since the pilot is actually part of the overall controller his precise role, the variables that are displayed to him, and the controls he uses to zero the display must be considered. Alternative roles for the pilot in the overall controller include: 1) using the pilot only in the loop, nulling the displays driven by the

trajectory controller, 2) giving the pilot vernier controls to refine the performance of the autopilot, or 3) flying the maneuver automatically after the pilot has achieved the appropriate inital conditions.

The first role is desirable since the pilot's isolation of faults and corrective action is much more reliable than that of the autopilot. Only this first role, with the pilot fully in the feedback control loop will be used in this study.

Besides the pilot role, the control design must also address which error signals should be displayed and which controls are most effective for a given trajectory. To simplify the pilot task the errors displayed should be dynamically decoupled with respect to the controls as much as possible, and also separated in frequency where possible. This enables the pilot to more easily coordinate and sequentially correct the trajectory errors.

There are two other aspects of the pilot task that effect the controller design, the pilot's preference for the "plant controller" dynamics, which is the combined plant that he controls, and the pilot's preference for the speed of the closed-loop system; i.e., the speed at which he prefers to control. We want to optimize the pilot performance by giving him a system which he can best control. Since the pilot responds differently to different systems, iterative controller design could be used to converge on the combined pilot controller-plant model. For this study, a published model of the human operator controlling a $1/s^2$ plant [1] will be used. Further research is currently being conducted, employing improved representations of pilot behavior using frequency shaping methods. Since the human operator adapts to the given system the $1/s^2$ model will be a conservative description of the pilot's actual performance in flight tests.

Pilot Model

A pilot model presented in previous studies has been

$$P(s) = P_{EQ}(s) \frac{e^{-\tau s}}{\frac{1}{\tau_N} s + 1} \quad , \tag{3.6}$$

where $P_{EQ}(s)$ is the human equalization network, τ_N is a neurological motor delay, and τ an observation delay. Figure 3-4 shows what has been called the control theoretic model of optimal human behavior.

Pilot Modeling Difficulties

The above control theoretic model requires the propagation of delay-differential equations which introduce considerable complication.

Between the two delays and the control and filter weights in the optimal regulator and estimator, there is considerable flexibility in this model to fit empirical data. If the system is driven with sinusoidal inputs then the assumption that the human can perfectly compensate for a delay will be self-fulfilling as the human operator settles into a steady-state after observing several cycles. Time domain approaches which include random inputs with a band of frequency content would be more desirable and should indicate a much lower gain at high frequencies, rather than primarily the large phase lag. Finally, the model is valid for a particular plant, in our case $1/s^2$, when the dynamics are actually higher order and not a pure integrator in the mid-frequency range where the pilot responds well.

The significant aspect of the human pilot, however, is his ability to identify and adapt to the plant dynamics, therefore our assumption of a $1/s^2$ plant is a conservative one for the overall trajectory controller and pilot system. Thus, the given $1/s^2$ model will suit our purpose in developing trajectory controller design techniques. The frequency response shown in Ref. [1] is given in Figure 3-5.

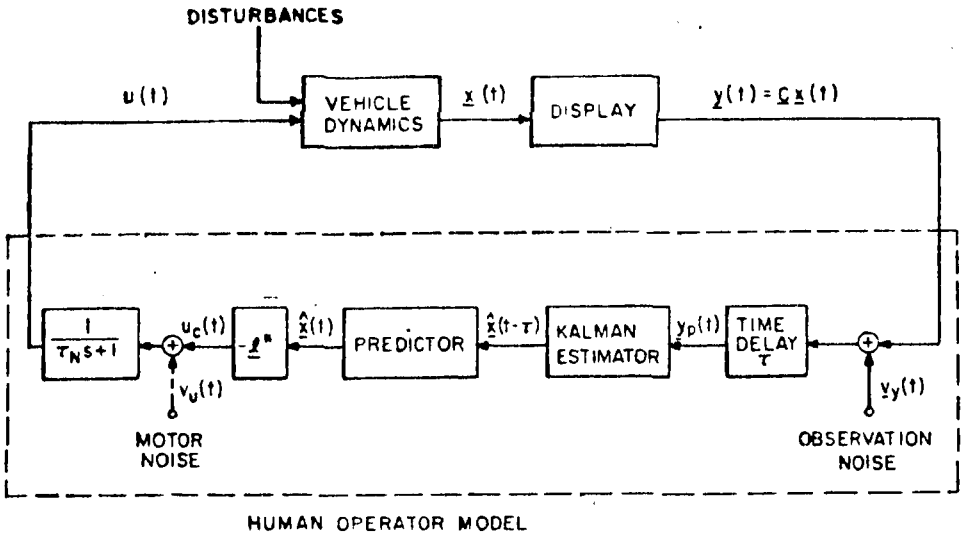

Figure 3-4 Control-Theoretic Model of Optimal Human Behavior

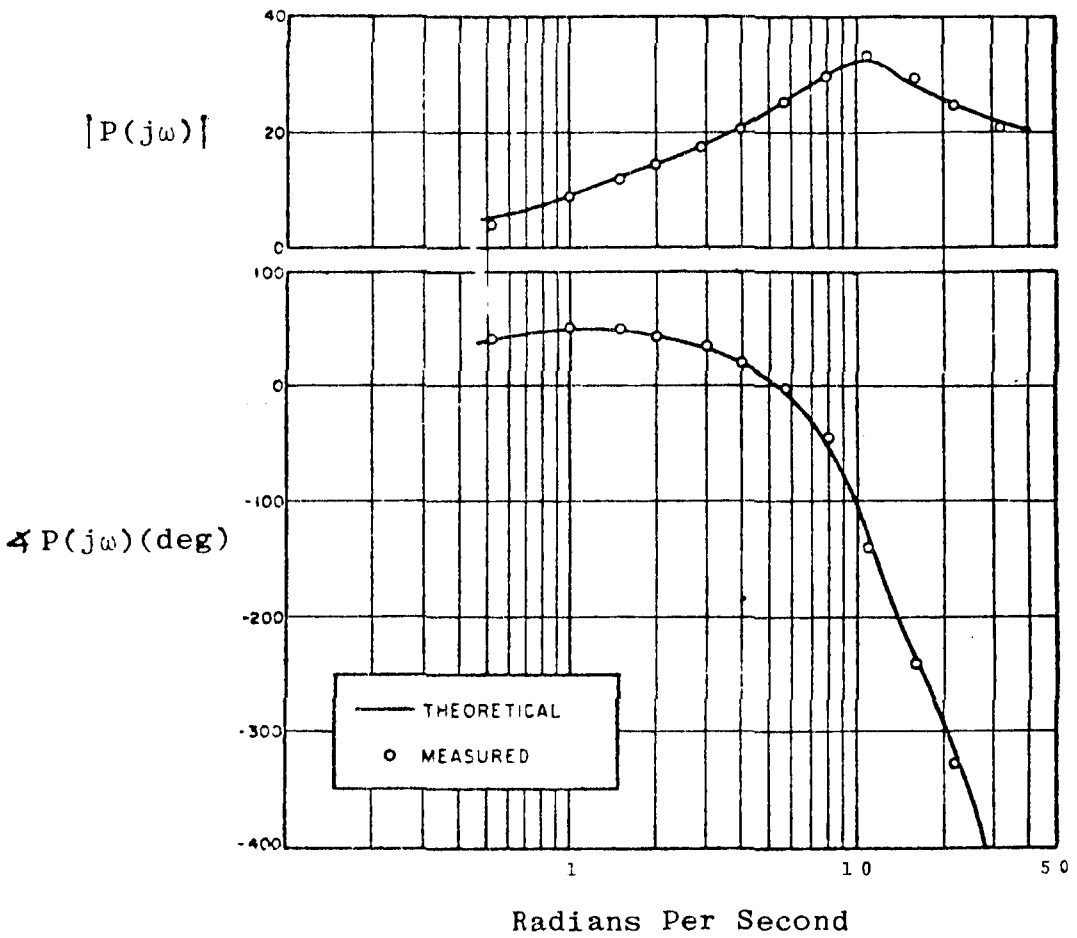

Figure 3-5 Pilot Transfer Function Controlling a $1/s^2$ Plant

The pilot limitation parameters are given in the following table:

Motor Delay	τ_N	100 msec
Observation Delay	τ	210 msec

The pilot transfer function is

$$P(s) = \frac{P_{EQ}(s)\ (9.52 - s)}{(s + 10)(s + 9.52)} \tag{3.7}$$

A reasonable linear approximation to the above frequency response is

$$P(s) = \frac{-56.1(s + .6)(s + 4.75)(s - 9.52)}{(s + 8)(s + 10)(s + 14)(s + 9.52)} \tag{3.8}$$

A state-space realization of this transfer function (using an observability form) is

$$P(s) = H_p\,[sI - F_p]^{-1}\,G_p\ , \quad \text{with} \tag{3.9}$$

$$H_p = [-56.1 \quad 0 \quad 0 \quad 0]$$

$$Fp = \begin{bmatrix} -10. & -356.98 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1066.7 & 321.53 & 31.524 \end{bmatrix}, \quad G_p = \begin{bmatrix} 1 \\ .1 \\ -2.1170 \\ 37.646 \end{bmatrix}$$

3.6 SUMMARY

A state space linear perturbation model for the overall system shown in Figure 3-1 appears as

(3.10)

$$\begin{bmatrix} \dot{x}_L \\ \dot{x}_A \\ \dot{x}_{MC} \\ \dot{x}_E \\ \dot{x}_p \\ \dot{x}_s \end{bmatrix} = \begin{bmatrix} F_L & F_{LA} & 0 & 0 & 0 & 0 \\ 0 & F_A & F_{AM} & F_{AE} & 0 & 0 \\ 0 & 0 & F_{MC} & 0 & F_{MP} & F_{MS} \\ 0 & 0 & 0 & F_E & F_{EP} & F_{ES} \\ 0 & 0 & 0 & 0 & F_p & 0 \\ 0 & 0 & 0 & 0 & 0 & F_s \end{bmatrix} \begin{bmatrix} x_L \\ x_A \\ x_{MC} \\ x_E \\ x_p \\ x_s \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ G_p \\ 0 \end{bmatrix} \begin{bmatrix} e_{D_V} \\ e_{D_H} \\ e_{D_M} \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma_A \\ 0 \\ 0 \\ \Gamma_p \\ \Gamma_s \end{bmatrix} \begin{bmatrix} w \\ n_p \\ n_s \end{bmatrix}$$

where the states, controls, and noise inputs are:

x_L = location coordinates (x, y, z)

x_A = aerodynamic/kinematic states

x_{MC} = MCS/CAS states

x_E = engine states

x_p = pilot states

x_s = sensor states,

e_{D_V} = vertical display error

e_{D_H} = horizontal display error

e_{D_M} = Mach display error

w = gust inputs

n_s = sensor noise

n_p = pilot motor noise

SECTION 4

CONTROL DESIGN PROCEDURE

The overall control-design procedure consists of three major steps:

1. Model reduction: Development of control-design models from detailed models of the previous section.

2. Control design: Design of a controller/filter or output-feedback logic to specify the control structure.

3. Controller simplification: Reduction of the controller obtained in the previous step to simplify implementation and for more robustness.

The steps in the overall procedure must be carefully integrated to support assumptions and approximations made in each of the steps. The development of specific design methodologies which support this integration is the basis of our control design methodology for flight test trajectory controllers.

We discuss each of these steps in some detail.

4.1 MODEL REDUCTION

Model reduction is required for robustness and to simplify control law computation and implementation.

Criteria and computation procedures for model reduction are discussed in this section. The reduced model must contain essential elements of the dynamics such that the resulting control law meets the design requirements. Roles of poles, zeros and residues of transfer functions in closed-loop control design are discussed first followed by selection of specific criteria and numerical procedures.

Poles, Zeros and Residues: In linear systems, the closed-loop behavior is dictated significantly by several transfer function properties. Three of the most significant are the poles, the zeros, and the residues.

Consider a linear model with nx1 state vector x, qx1 control vector u, and px1 output vector y:

$$\dot{x} = Fx + Gu \ , \tag{4.1}$$

$$y = Hx + Du \ . \tag{4.2}$$

The transfer function between y and u is

$$\begin{aligned} y(s) &= [H(sI-F)^{-1}G + D]\, u(s) \\ &\triangleq T(s)\, u(s) \ . \end{aligned} \tag{4.3}$$

The roles of transfer function properties are best explained by considering a single-input single-output (SISO) system

$$y(s) = T(s)\, u(s) \tag{4.4}$$

The characteristic values λ_i, $i = 1, 2, \ldots, n$ of the $T(s)$ denominator are the open loop system eigenvalues and indicate system stability properties. The transfer function may be written in terms of zeros z_i, $i = 1, 2, \ldots, n$ or residues r_i, $i = 1, 2, \ldots, n$ as follows:

$$T(s) = \frac{K(s-z_1)(s-z_2) \ldots (s-z_m)}{(s-\lambda_1)(s-\lambda_2) \ldots (s-\lambda_n)} \ : \text{Zeros Representation} \tag{4.5}$$

$$= \sum_{i=1}^{n} \frac{R_i}{s-\lambda_i} \qquad : \text{Residue Representation} \tag{4.6}$$

Consider now a feedback matrix $C(s)$ with gain α. The closed-loop transfer function is

$$T_c(s) = \frac{T(s)}{1 + \alpha C(s)\, T(s)} \tag{4.7}$$

which may be written as

$$T_c(s) = \frac{\bar{K}(s-\bar{z}_1) \ldots (s-\bar{z}_{\bar{m}})}{(s-\bar{\lambda}_1)(s-\bar{\lambda}_2) \ldots (s-\bar{\lambda}_{\bar{n}})} \tag{4.8}$$

It is easy to see that for smaller α,

$$\left.\frac{\partial \lambda_i}{\partial \alpha}\right|_{\alpha \to 0} = \alpha C(\lambda_i)\, r_i \tag{4.9}$$

Thus r_i dictates the behavior of the pole for small gain. When α is large, the finite closed-loop poles are the zeros of $C(s)$ and $T(s)$.

To summarize, the residues of the transfer function describe the low-gain properties and the zeros the high-gain properties. Both zeros and residues are important in closed-loop control design. Therefore, an ideal reduced-order model should maintain the residues of the retained poles and zeros in the spectrum of interest. Unfortunately, both zeros and residues cannot be preserved simultaneously. The attempt in reduced-order modeling methods is to maintain either poles or zeros or provide approximations to both of them.

Retention of Residues

The procedure for retaining residues is implemented as follows. Let F, G, H and D be in modal form and assume that the first n modes are retained. The state equations are then written as

$$\begin{bmatrix} \dot{x}_1 \\ \hline \dot{x}_2 \end{bmatrix} = \left[\begin{array}{c|c} F_{11} & 0 \\ \hline 0 & F_{22} \end{array}\right] \begin{bmatrix} x_1 \\ \hline x_2 \end{bmatrix} + \begin{bmatrix} G_1 \\ \hline G_2 \end{bmatrix} u \,, \tag{4.10}$$

$$y = [H_1 \mid H_2] \begin{bmatrix} x_1 \\ --- \\ x_2 \end{bmatrix} + Du \ . \qquad (4.11)$$

Let the average frequency of the retained model be ω_a. The state x_2 is approximated by

$$x_2 = \mathrm{Re}[(j\omega_a I - F_{22})^{-1} G_2]\, u \ , \qquad (4.12)$$

where $\mathrm{Re}(\cdot)$ represents the real part of $(\cdot)$. Therefore, the reduced model is approximated by

$$\dot{x}_1 = F_{11}x_1 + Gu \ ,$$

$$y = H_1 x_1 + [D + \mathrm{Re}[H_2(j\omega_a I - F_{22})^{-1} G_2]]\, u \ . \qquad (4.13)$$

If we are interested in matching the dc gain of the reduced model to that of the high-order model, ω_a may be set to zero. Note that only the real part of the term in Eq. 4.12 is retained because we want reduced-order models with real coefficients.

Retention of Zeros

Suppose we want to retain the first m zeros and n poles. Let ω_a be the average frequency. The transfer function of Eq. 4.5 is simplified to

$$T_{k\ell}(s) = \frac{g(j\omega_a - z_{m+1}) \ \ldots \ (j\omega_a - z_M)}{(j\omega_a - \lambda_{n+1}) \ \ldots \ (j\omega_a - \lambda_N)} \ \frac{(s-z_1) \ \ldots \ (s-z_m)}{(s-\lambda_1) \ \ldots \ (s-\lambda_n)} \ . \qquad (4.14)$$

The first term is approximated by a real gain. Each input-output transfer function is simplified as in Eq. 4.14. These simplified transfer functions are reconverted into a state-variable description.

The computation time requirements to obtain reduced-order models which retain zeros are very high. Poles and residues of high-order models are determined much more easily than zeros. This model-reduction procedure, therefore, has been used rarely. Methods that maintain system transmission zeros have been described using zero directions [2]; however, a systematic design method employing MIMO zeros has not yet been developed. The transmission zeros are the frequencies at which there is no output to any combination of inputs. They are the eigenvalues of the generalized eigenvalue problem

$$\begin{bmatrix} F & G \\ H & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} = z \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{4.15}$$

Mode Detection Criteria

If the model is to be reduced to n' states, the number of poles will drop to n'. Criteria have to be developed to determine the set of poles which may be dropped, along with the corresponding set of modes. These criteria must consider the following:

1. Any mode which is uncontrollable and undisturbable or unobservable can be dropped. Either of these conditions corresponds to a zero residue or a perfect pole-zero cancellation in all transfer functions.

2. Controllable modes should not necessarily be discarded even if they are nondisturbable and add nothing to the cost functional, since they may be excited by the control actuators.

3. Highly controllable modes in the bandwidth of interest should be retained even if they are not disturbable or observable. Similarly, highly observable modes should be retained for robustness reasons.

4. Proper mode ordering based on cost analysis [3] should include the above factors in addition to performance considerations.

It is clear that many simplifications of the model form are possible. In general, the analyst would start from the simplest model and carry the procedure through to evaluation. If the performance is unsatisfactory, a more detailed model should be considered.

4.2 SELECTION OF CONTROL LAW AND STATE ESTIMATOR

Multivariable linear-quadratic-regulator (LQR) control theory is a powerful tool for the development of flight test trajectory controllers. If the measurements have unacceptable noise or if unmeasured states are needed for feedback, a state estimator may also be designed to supplement the regulator.

4.2.1 Basic Linear Quadratic Regulator Approach

The basic approach is described first. Extensions of the approach for the test trajectory control problem will then be covered. Consider a linear system model in which the states x and inputs u represent perturbation around nominal values x_o and u_o

$$\dot{x} = Fx + Gu \ . \tag{4.16}$$

In this problem, u represents the variables to be displayed (nominally roll, pitch and Mach number errors) and x depends on the particular model dynamics selected (u, w, z, T and h_m for the simplest model). The control design problem selects u to drive the perturbation states x to zero. The LQR formulation optimizes a quadratic performance index of the form

$$J = \frac{1}{2}\int_0^T (x^TAx + u^TBu)\, dt \ . \tag{4.17}$$

The control law is obtained by solving the Riccati equation

$$\dot{S} = - SF - F^TS - A + SGB^{-1}G^TS$$

and

$$u(t) = - B^{-1}G^{T}S\, x(t) = C(t)\, x(t) . \qquad (4.18)$$

To simplify implementation, a steady-state solution could be used where the $\dot{S}$ is set to zero and a constant value of the gain matrix $C(t)$ is computed. This constant control gain can be used throughout the maneuver. Note that since x and u are perturbation values, the steady state x_o and u_o must be subtracted from the measured quantities prior to computing the control input, and is added into the final control applied.

Each maneuver is specified in terms of desired values of the 12 aircraft state variables or functions of these variables (the 12 variables are the three components each of position, velocity, Euler angles and angular rates.) The specification can be in one of the following forms for each state (or function of states):

1. desired constant value,
2. desired constant value for integral or derivative,
3. desired time history, or
4. desired value at some point on the trajectory.

4.2.2 Constant Values

A desired constant value is equivalent to having trim values as discussed above. The steady-state states and controls necessary to achieve the desired outputs can be solved from the equation

$$\begin{bmatrix} x_o \\ u_o \end{bmatrix} = \begin{bmatrix} F & G \\ H & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ z_c \end{bmatrix} \qquad (4.19)$$

If the system has open-loop zeros at the origin, this inverse does not exist, or in other words, there is no neighboring steady-state value. In this case integral error states can be

added to achieve the desired constant value. Constant integral as well as derivative states are discussed next.

4.2.3 Constant Integral or Derivative Regulation

Integral control: To account for errors in computation of trim conditions and other low-frequency modeling errors, it is necessary to place a penalty on the integrals of the error signals. For example, if a state z is being held at z_o, we can define

$$\dot{\xi} = z - z_o \tag{4.20}$$

and place an additional penalty on ξ as follows:

$$\xi^T A_\xi \xi \quad . \tag{4.21}$$

Regulation of derivatives of states: To control derivatives of states, additional terms are added to the performance index. For example, to make $E\dot{x}$ follow E_o, the following term is added to the integrand of J:

$$(E\dot{x} - E_o)^T A_e (E\dot{x} - E_o) \ ,$$

or

$$[E(Fx + Gu) - E_o]^T A_e [E(Fx + Gu) - E_o] \ . \tag{4.22}$$

The control laws can be computed directly for the new performance index and will be of the form:

$$u(t) = - Cx(t) + C_e E_o \ . \tag{4.23}$$

(References [4] and [5] describe a new stable and reliable algorithm for solving the algebraic Riccati equation with cross-coupling between state and control cost.)

Integral control is valuable for zero steady-state errors although it can lead to a sluggish controller if the feedback consists primarily of the integral state, since it takes time for the error to build up.

Derivative control on the other hand gives excellent response, and in fact, is called dynamics matching in the context of implicit model following. A typical trajectory where such a control law would be possible is the specification of loads, where linear combinations of the derivatives of rate equations are to be regulated about a constant value.

4.2.4 Controlling the Time History of a State Variable

The technique can be extended to make certain variables (say Tx) follow a time history T(t). The integrand of the performance index is extended as follows:

$$(Tx - T(t))^T A_t (Tx - T(t)) \ . \tag{4.24}$$

The control law now requires the solution of a differential equation backward in time.

The complete problem formulation as described in Ref. [6] is

$$J = \frac{1}{2} [z_0 - H_0 x(t_0)\|^2_{A_0} + \frac{1}{2} \int_{t_0}^{t_f} [\| z - Hx \|^2_{A^{-1}} + \| u \|^2_{B^{-1}}]dt + \frac{1}{2} \| z_f - H_f x(t_f) \|^2_{A_f^{-1}} \tag{4.25}$$

where

$$\dot{x} = Fx + Gu \ , \tag{4.26}$$

given z_0, H_0, A_0, $z(t)$, $H(t)$, $A(t)$, $B(t)$, z_f, H_f, A_f, $F(t)$, $G(t)$, t_0, and t_f.

$x(t)$ is the state vector of a dynamic system.

$H(t)\,x(t)$ is a vector output history.

$z(t)$ is a desired vector output history (prespecified).

$u(t)$ is a vector control input history.

$H_0 x(t_0)$ and $H_f x(t_f)$ are initial and final vector outputs.

z_0 and z_f are desired initial and final vector outputs.

A_0, A_f, $A(t)$, and $B(t)$ are relative weighting matrices to be specified by the control designer.

The desirable solution for this "follower" problem is a backward sweep solution derived from the Euler-Lagrange equations, of the form

$$\lambda(t) = -\lambda_B(t) + S_B(t)\,x(t) \,. \tag{4.27}$$

Differentiating (4.27) with respect to time and using the Euler Lagrange equations yields the backward sweep equations:

$$-\dot{S}_B = S_B F + F^T S_B - S_B G_B^{-1} G^T S_B + H^T A H \,,$$

$$S_B(t_f) = H_f^T A_f H_f \,, \tag{4.28}$$

$$-\dot{\lambda}_B = (F - GC_B)^T \lambda_B + H^T A z \,,$$

$$\lambda_B(t_f) = H_f^T A_f z_f \,, \tag{4.29}$$

where

$$C_B \triangleq B^{-1} G^T S_B \,. \tag{4.30}$$

Equation (4.28) is a Riccati equation for the symmetric matrix S_B. It must be integrated backward from $t = t_f$ to $t = t_0$ along with (4.29) for the vector λ_B.

If $C_B(t)$ and $B^{-1}G^T\lambda_B(t)$ are stored on the backward sweep, then $\dot{x}$ and $\dot{\lambda}$ can be integrated forward to determine $u(t)$:

$$\dot{x} = Fx + Gu \ , \tag{4.31}$$

where

$$u = u_B - C_B x \ , \tag{4.32}$$

$$u_B = B^{-1}G^T\lambda_B \ . \tag{4.33}$$

Note that this set of equations involves linear feedback of the state vector $x(t)$. Equation (4.32) is a feedback plus feedforward from of solution.

For linear time-invariant systems with which we have approximated our system, the Riccati equation need not be integrated but rather a vector forcing function using model decomposition for the linear two point boundary value problem (TPBVP) (see [7] for the distinct eigenvalue case and [8] for the general eigenstructure case). However, except for 4-D guidance, we do not need to control with a finite time formulation, but rather can employ steady-state control laws. This is equivalent to saying that $z(t_f)$ persists as a constant value for a long period.

This further simplification gives a constant gain solution for the control

$$u = B^{-1}G^T\lambda_B - B^{-1}G^TS_Bx = u_B - C_Bx \ , \text{ and} \tag{4.34}$$

$$S_B(t) = S_B = \Lambda_- X_-^{-1} \quad \text{where} \begin{bmatrix} X_- \\ \cdots \\ \Lambda_- \end{bmatrix} \tag{4.35}$$

is an orthogonal basis for the stable eigenvalue of the Euler-Lagrange equations derived from the Hamiltonian.

Three approaches to solving these equations without integrating $\lambda_B(t)$ backward in time are:

1. Take $\dot{\lambda}_B \approx 0$ which implies that $\lambda_B \approx -F_c^{-1}H^TAz$, so that $u = u_B - C_Bx$, can be computed from z and x.

2. Consider $z(t)$ as a constant, so that $u = u_0 - C_B(x - x_0)$.

3. Approximate $z(t)$ as a function of the states

   a. A linearization of a nonlinear relationship between the states with integral error to correct the linearization.

   b. A random walk which can be integrated forward in time with $\dot{x}$.

1. and 2. above are equivalent, both relying on the assumption that the rate of change of $z(t)$ is much less than the slowest eigenvalue of $F_c = (F - GC_B)^T$. This is a reasonable assumption for most trajectories, since commanded outputs are not rapidly changing and the controllable roots of F_c can be made faster with feedback.

To approximate $z(t)$ as a function of the states, the first alternative is not particularily attractive since an accurate linearization may not exist and the integral error tends to slow down the response and may also constrain the trajectory unnecessarily. The random walk is a much more desirable way to generate a linear trajectory. The random walk also is affected by the rate of change of $z(t)$ but not to the same extent as considering $dz/dt = 0$ or $\dot{\lambda}_B = 0$. The overall variance of the error in y may be comparable to the other approaches, but the model can be "tuned" to give a smaller spectral density in the range of interest.

Changing the direction of integration in Equation (4.29) gives

$$\dot{\lambda}_B = (F - GC_B)^T\lambda_B + H^TAz(t) , \qquad (4.36)$$

with an <u>unknown</u> initial condition.

The appropriate initial condition $\lambda_B(t_0)$ appears as an impulsive input on the above equation. This is true at any

initial condition as well, hence we can merely append a white noise source to equation (4.36), an additive random walk term

$$\dot{\lambda}_n = (F - GC_B)^T \lambda_n + H^T Az(t) + n_\lambda \ , \quad \text{with}$$

$$\lambda_n(t_0) = 0 \ , \tag{4.37}$$

$$n_\lambda = N(0, Q_n) \ ,$$

and Q_n can be chosen to bracket the desired value for $\lambda_B(t_\infty)$. Since

$$\lambda_B(t_\infty) = H^T Az(t_\infty) \ ,$$

while Eq. (4.37) gives

$$\lambda_n(t_\infty) = -F_c^{-1} H^T Az(t_\infty) \ ,$$

and Eq. (4.37) implies

$$0 = F_c \tilde{\Lambda}_n + \tilde{\Lambda}_n F_c^T + Q_n \Rightarrow \tilde{\Lambda}_n \ , \tag{4.38}$$

The RMS values from $\tilde{\Lambda}_n$ give the interval about $\lambda_n(t_\infty)$ which should bracket the theoretical steady-state value $\lambda_B(t_\infty)$ with an acceptable level of confidence or integral error states can be added for λ_n.

More simply, the concept of the costate, λ_B, can be dropped by using several appended random walk states driven by $(Hx(t) - z(t))$.

4.2.5 Controlling a Set of States at Some Point on the Trajectory

There are several ways to solve this problem. One approach is to select a state time history which leads to the desired

trajectory point and then to derive a control law which follows the time history. It is also possible to define the trajectory point as a function of several states and then to hold this desired state fixed while other states are varied. The particular selection will depend on the desirable nature of the intermediate trajectory.

A combination of the above controllers can be used, if necessary.

4.2.6 Extensions and Specializations

Several extensions are proposed to improve the quality of the control laws. These extensions are designed to minimize the effects of modeling errors, lack of needed measurements and noise:

1. Estimator: If all states are not measured or if there is excessive random or quantization noise, v, a state estimator can be used. If y is a vector of measurements which is linear in the states, then

$$y = Hx + v \ . \tag{4.39}$$

An estimate of the state can be obtained by

$$\dot{x} = Fx + Gu + K(y - Hx) \ , \tag{4.40}$$

and K is obtained by solving a Riccati equation. (See the next section on controller simplification for a desirable algorithm in the case where some but not all of the measurements are noise free.)

2. Frequency-shaped control law: As shown in Appendix A, the control weightings can be made functions of frequency. By making the control weighting matrix an increasing function of frequency and the state weighting a decreasing function of frequency, the control activity can be concentrated in the mid-frequency range. The display will then have a minimum of high-frequency components. This shaping will also reduce the effect of neglected states with high natural frequency [9].

4.3 CONTROLLER SIMPLIFICATIONS

The controllers based on the LQR approach are simple and usually do not need to be simplified. LQG controllers involve a dynamic state estimator and are generally of the form

$$\dot{x}_c = F_c x_c + K_y ,$$

$$u_c = C x_c + D_c y . \qquad (4.41)$$

The controller, therefore, has the same form as the system itself, except that: (1) the roles of input and output are switched, (2) the controller state vector depends on the choise of frequency shaping, and (3) K and C are known exactly (unlike G and H in the design model). Controller simplification is, therefore, similar to model reduction (see Section 4.1).

Noiseless Measurements and Reduced Order Compensators

The combination of the regulator and estimator can be viewed as a compensator for the open loop plant, where the compensator transfer functions are given by

$$u(s) = T_c(s)\, z(s) ,$$

$$u(s) = - C(sI - F + FC + KH)^{-1} Kz(s) \qquad (4.42)$$

This transfer function can be written in modal form as a parallel bank of first and second order equations.

If some of the measurements are free of white noise, a reduced order compensator can be determined. Partitioning the measurements sc that the white-noise-free ones are at the bottom, $z^T = \left[z_1^T \;\vdots\; z_2^T \right]$, the computed estimator gains are $K = [K_1 \;\vdots\; 0]$.

These gains result from the reduced order Riccati solution

$$P = \begin{bmatrix} P_{11} & 0 \\ 0 & 0 \end{bmatrix} , \qquad (4.43)$$

which can be solved for directly with recent stable and reliable algorithms [4,5].

The compensator feed-through term, D_c, where

$$u(s) = [-C(sI - F + GC + KH)^{-1} K + D_c] Z(x) , \qquad (4.44)$$

can be computed from z_2 which has no error, by considering that

$$\begin{bmatrix} z_1 \\ \hline z_2 \end{bmatrix} = \begin{bmatrix} H_1 \\ \hline H_2 \end{bmatrix} \hat{x} + v , \quad \text{and} \quad u = -C\hat{x} \qquad (4.45)$$

The requested feedback gains are decomposed into

$$C = C_D + C_y ,$$

with C_D that portion explained by the noise-free measurements $z_2 = H_2 x$. C_D is the intersection of the row space of H_2 with the row space of C. If the intersection is zero there is no feed through, if the intersection is all of C then the dynamic compensation portion C_y is zero. The feed-forward portion,

$$u(s)_{\text{feedthrough}} = -C_D H_2^T z_2 = [\ 0 \ \vdots \ D_{c_2}\ ]\ z \ , \qquad (4.46)$$

is solved from the linear equations,

$$H_2^T D_{c_2}^T = C_D^T \ . \qquad (4.47)$$

All unobservable or uncontrollable modes of the controller should be dropped, since they have no effect on controller performance. In addition, poles with small residues can also be dropped. Note that the residues are physical matrices representing force or moment applied per unit error.

Further selection of modes or states which should be retained in the low-order controller is a difficult problem. One approach is to drop one or two states at a time to determine the set of states which may be eliminated without loss of performance. To minimize computation time, a stepwise search procedure is needed. A modal cost function could be used in the search [10].

SECTION 5

ZOOM AND PUSHOVER ALGORITHM DEVELOPMENT

This section describes the maneuver modeling, development of a reduced order linear design model, control design results, their evaluation on the full order linear evaluation model and demonstration in the nonlinear simulation. The next section documents how this flight test trajectory controller can be implemented for the zoom and pushover maneuver.

5.1 TRAJECTORY MODELING

The zoom and pushover trajectory has been designed to maximize the time at the target angle-of-attack, or alternatively maintain a constant vertical acceleration (with zero horizontal acceleration).

A parabolic flight path with apex at z_o has the property that

$$\tan \gamma = \tan(\theta - \alpha) = \xi(z - z_o)^{1/2} , \qquad (5.1)$$

where γ is the flight path angle and ξ is a constant of proportionality, which defines the size of the parabola. An approximate model can be written as

$$\theta - \alpha = \xi(z - z_o)^{1/2} , \quad (\theta-\alpha) << 90^\circ . \qquad (5.2)$$

In addition, the following conditions must hold

$$\phi = 0 . \qquad (5.3)$$

M and α are given at the apex. Our controller design will try to achieve the speed time history of a ballistic trajectory, such that the thrust nominally balances drag. With such a time history, the thrust would be automatically stabilized for the target angle-of-attack with desired Mach number and altitude at the apex of the parabola. Thus, the Mach number at any altitude is given by:

$$M^2 - M_o^2 = 2g(z - z_o)/c^2, \tag{5.4}$$

where c is the speed of sound which is assumed constant on the trajectory (a variable value can also be accommodated). The equation can be linearized as

$$2(M - M_o)M_o = \frac{2g}{c^2}(z - z_o) , \tag{5.5}$$

or

$$M - M_o = \frac{g}{M_o c^2}(z - z_o) .$$

Note that the correct value of the speed will be reached at the apex because when $z = z_o$, M will be forced to M_o.

ξ is selected such that the lift defined by Mach number and angle-of-attack at the apex exactly balances gravity and the centrifugal force. A large value of ξ will be needed when the target angle-of-attack or Mach number is small and vice versa (because small lift will require a tighter curve). It can be shown that with vertical acceleration a_z and speed V at apex, ξ can be selected as:

$$\frac{V^2 \xi^2}{2} = g - a_z \tag{5.6}$$

since the radius of curvature at the apex is $2/\xi^2$.

Zoom and pushover is defined with θ-α and M as functions of altitude and the bank angle held at zero. θ-α and M time histories are two of several outputs of interest

$$y = [a_z,\ \alpha,\ q,\ \gamma,\ M] ,$$

which can be generated from a linear model as a function of time (see section 3).

Since the thrust is stabilized, the exact final altitude cannot be achieved in the presence of gusts, hence we program $\gamma_c(t)$ and $M_c(t)$ as functions of <u>time</u> rather than <u>altitude</u>. On a parabolic trajectory $(z - z_o)^{1/2}$ is linear in time. Since the trajectory command is open loop (due to the stabilized thrust constraint), the nonlinear relation between $\gamma(t)$ and M(t) can be used as shown below. The commanded output is then

$$y_c(t) = \begin{bmatrix} a_{z_c}(t) \\ \alpha_c(t) \\ q_c(t) \\ \gamma_c(t) \\ M_c(t) \\ \end{bmatrix} = \begin{bmatrix} a_o \\ \alpha_c(a_{z_c}, \gamma_c, M, q_\infty) \\ 0 \\ -\gamma_i/t_f + \gamma_i \\ \left[\frac{gR_o}{c^2}[\gamma_c(t)]^2 + M_o\right]^{1/2} \end{bmatrix} = \begin{bmatrix} p_1 \\ - \\ p_3 \\ \gamma_i t/p_6 - \gamma_i \\ (p_7\gamma_c^2(t) + p_5^2)^{1/2} \end{bmatrix}$$

The angle-of-attack is not arbitrary since it is determined by the commanded a_{z_c} and γ_c and the actual Mach and dynamic pressure. The flight path angle is programmed from an initial value to zero at the apex.

A summary of the zoom and pushover maneuver algorithm development is:

1) Choose α_o, a desired apex angle-at-attack, and z_o, the desired apex altitude.

(2) Trim the aircraft at the apex altitude and angle of attack.

(3) Find a final Mach number M_o (which must be slower than the trimmed value to be on a parabola at the apex). This value cannot be arbitrary since the lift and drag are also functions of velocity. The force balance equations are:

$$\frac{mV^2}{R_o} = - L - T \sin \alpha + W ,$$

$$0 = - D + T \cos \alpha$$

We seek the apex velocity and thrust from

$$\begin{bmatrix} \frac{+ c_L \rho s}{2} + \frac{m}{R_o}, & + \sin \alpha \\ \frac{- c_D \rho s}{2}, & \cos \alpha \end{bmatrix} \begin{bmatrix} V_o^2 \\ T_o \end{bmatrix} = \begin{bmatrix} W \\ 0 \end{bmatrix}$$

The acceleration V_o^2/R_o specifies the shape of the parabola and hence the constants in $\gamma_c(t)$ and $M_c(t)$.

(4) Chose either the initial altitude or the time of flight (within an altitude range of approximately constant speed of sound, one specifies the other).

(5) Find the initial flight path angle and Mach number for the desired initial altitude, z_i, from

$$M_i = \left[\frac{2g}{c^2} (z_o - z_i) + M_o^2 \right]^{\frac{1}{2}} ,$$

$$\tan \gamma_i = \xi (z_o - z_i)^{\frac{1}{2}}$$

(6) Regulate $(a_z - a_{z_o})$, $(\alpha - \alpha_c(t))$, q, $(\gamma - \gamma_c(t))$, and $(M - M_c(t))$ about zero with a robust linear controller.

5.2 ZOOM AND PUSHOVER (ZAPO) LINEAR MODELS

This subsection describes linear models for: 1) the F-15 aerodynamics, MCS/CAS, and sensor dynamics; and 2) the piloted aircraft with reduced order models for both.

The trimmed condition of the F-15 used in the simulation SIMII is shown in Table 5-1.

Altitude	h_o	32,000 ft.
Angle of Attack	α_o	11°
Pitch Angle	θ_o	11°
Mach Number	M_o	.4775
Velocity	V_o	470.9 ft/sec
Thrust	T_o	6270 lb_f
Weight	W_o	40,700 lb_f
Stabilator	δ_{SB_o}	-4.76°
Dynamic Force	$\bar{q}s$	55,821.1 lb_f

Table 5-1. ZAPO Trim Condition*

*Pitch, roll, and yaw CAS on, flaps up, with autopilots off.

The six nondimensional aerodynamic coefficients and their associated dimension stability derivatives reflected into the appropriate body fixed axis are given below in Table 5-2.

The MCS/CAS equations (ignoring servos at 30, 60, and 100 radians/sec) from Figure 3-3 are

$$\dot{\delta}_{SB_1} = -.33\delta_{SB_1} + 15.5q + 15.5\dot{q}$$

$$\dot{\delta}_{SB_2} = -5.\ \delta_{SB_2} + 11.49\ \delta_e$$

$$\dot{\delta}_{SB_3} = .882\dot{q} + .4\ a_z$$

$$= .882\dot{q} + 5.85\ (\dot{\alpha} - q)$$

$$\dot{\delta}_{SB} = -20.\delta_{SB} + 20/57.3(\delta_{SB_1} + \delta_{SB_2} + \delta_{SB_3}$$

$$+ .6832\dot{q} - 4.476\delta_e)\ .$$

The sensor and thrust equations are

$$\delta M = -.1\delta M + .1\left(\frac{\delta V}{C}\right)$$

$$\left(\frac{\delta T}{M}\right) = -.1\left(\frac{\delta T}{M}\right) + .1\left(\frac{\delta T_c}{M}\right)$$

	NONDIMENSIONAL CL,CM,CN,CD,CLFT,CY = -3.15007E-13 -1.28433E-02 1.24240E-06 1.10237E-01 7.05545E-01 4.44849E-12

	CLBODY	CM	CNBODY	CD	CLFT	CYBODY
Angle of attack	ALPH -0.36707E-02	-0.15745E+01	-0.22784E-03	-0.61949E+02	-0.16598E+03	0.15015E-02
Pitch rate	Q 0.00000E+00	-0.91294E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
Vertical velocity	W -0.76929E-05	-0.32819E-02	-0.47492E-06	-0.19277E+00	-0.31498E+00	0.31297E-05
Vertical acceleration	ANZ 0.00000E+00	-0.22116E-02	0.00000E+00	0.00000E+00	-0.30785E-01	0.00000E+00
Angle of attack rate	ALPD 0.00000E+00	-0.76451E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
Total velocity	VELO -0.35605E-06	0.23469E-03	-0.35863E-07	-0.18801E-02	-0.40951E-02	0.50076E-06
Velocity along A/C nose	U 0.14954E-05	0.63794E-03	0.92315E-07	0.37470E-01	0.61226E-01	-0.60835E-06
Elevator	DH 0.00000E+00	-0.38012E+01	-0.42944E-05	-0.12220E+02	-0.25475E+02	0.00000E+00
Sideslip	BETA -0.10814E+02	-0.30038E-06	0.10660E+01	0.00000E+00	0.00000E+00	-0.42589E+02
Roll rate	P -0.91636E+00	0.00000E+00	-0.90524E-02	0.00000E+00	0.00000E+00	0.00000E+00
Yaw rate	R 0.14781E+01	0.00000E+00	-0.25928E+00	0.00000E+00	0.00000E+00	0.00000E+00
Lateral velocity	V -0.83963E+01	-0.26201E-06	0.21168E+00	0.00000E+00	0.00000E+00	-0.37148E+02
Aileron	DA 0.25576E+01	0.74124E-09	0.10425E-01	0.00000E+00	0.00000E+00	0.10509E+00
Rudder	DR -0.23870E+00	-0.58042E-07	-0.10299E+01	0.00000E+00	0.00000E+00	-0.82293E+01
Differential rudder	DT 0.33509E+01	-0.27136E-07	0.45272E+00	0.00000E+00	0.00000E+00	-0.38474E+01

Table 5-2. F-15 Aerodynamic Model

The matrices of the linear model

$$\dot{x} = Fx + Gu + \Gamma w$$

$$y = H_y x \ , \quad z = H_z x + v$$

are given below where the states

$$x = \left[\delta V,\ \delta\alpha,\ q,\ \delta\theta,\ \delta_{SB_1},\ \delta_{SB_2},\ \delta_{SB_3},\ \delta_{SB},\ \delta M,\ \left(\frac{\delta T}{M}\right)\right]$$

are velocity, angle-of-attack, pitch rate, pitch, four MCS/CAS states, lagged Mach number, and lagged thrust. The inputs

$$u = \left[\delta_e \quad \delta T_{c/M}\right]^T \ , \quad w = \alpha \text{ gust},$$

$$\text{and } w = N \ , \quad \left(0,\ .1125^{\backslash -4}\ \text{rad}^2\text{-sec}\right)$$

are commanded elevator, thrust acceleration, and angle-of-attack gust. The outputs

$$y = \left[\delta a_z,\ \delta\alpha,\ q,\ \delta\gamma,\ \delta M\right]^T$$

are acceleration, angle of attack, pitch rate, flight path angle $(\delta\theta - \delta\alpha)$, and Mach number. The measurements

$$z = \left[\delta a_z,\ \delta\alpha,\ q,\ \delta\theta,\ \delta M\right]^T \ ,$$

are acceleration, angle-of-attack, pitch rate, pitch, and Mach number with

$$V = N\Big(0,\ \text{diag}\Big[.2588\ \text{ft}^2/\text{sec}^3,\ 7.62^{\backslash -6}\ \text{rad}^2\text{-sec}$$
$$1.755^{\backslash -6}\ \text{rad}^2/\text{sec},\ .1493^{\backslash -4}\ \text{rad}^2\text{-sec},\ 2.5^{\backslash -6}\text{-sec}\Big]\Big) \ .$$

The equations are scaled so that all angles, angular rates, and angular accelerations are in units of (.01) radians; velocities and

accelerations are in feet, with seconds the time unit. Mach number is scaled in units of (.001) Mach.

The linear system matrices are

F =

COLUMNS 1 THRU 8

-0.0019	-0.3072	0.0000	-0.3220	0.0000	0.0000	0.0000	-0.1222
-0.0009	-0.3420	1.0000	0.0000	0.0000	0.0000	0.0000	-0.0541
0.0235	-1.5480	-0.9890	0.0000	0.0000	0.0000	0.0000	-3.8000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.3643	-23.9940	0.1705	0.0000	-0.3330	0.0000	0.0000	-58.9000
0.0000	0.0000	0.0000	0.0000	0.0000	-5.0000	0.0000	0.0000
0.0156	-3.3660	-0.8723	0.0000	0.0000	0.0000	0.0000	-3.6681
0.0056	-0.3692	-0.2359	0.0000	0.3490	0.3490	0.3490	-20.9063
0.1014	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

COLUMNS 9 THRU 10

0.0000	0.9816
0.0000	0.0019
0.0000	-0.0001
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
-0.1000	0.0000
0.0000	-0.1000

G =

0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
11.4900	0.0000
0.0000	0.0000
-1.5620	0.0000
0.0000	0.0000
0.0000	0.1000

GM

GM =

-0.3072
-0.3420
-1.5480
0.0000
-23.9940
0.0000
-3.3660
-0.3692
0.0000
0.0000

HY =

COLUMNS 1 THRU 8

-0.0041	-1.6105	0.0000	0.0000	0.0000	0.0000	0.0000	-0.2548
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	-1.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

COLUMNS 9 THRU 10

0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
1.0000	0.0000

HZ =

COLUMNS 1 THRU 8

-0.0041	-1.6105	0.0000	0.0000	0.0000	0.0000	0.0000	-0.2548
0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

COLUMNS 9 THRU 10

0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
0.0000	0.0000
1.0000	0.0000

Table 5-3 shows the eigenvalues of the aerodynamics alone as well as the aerodynamics as modified by the MCS/CAS. For level flight (the final condition at the apex of the ZAPO maneuver) at such a high angle-of-attack, the MCS/CAS (without the autopilot) makes the phugoid mode unstable. The time constant is on the order of the maneuver flight time, however, and the flight test trajectory controller (FTTC) will compensate it as well.

Aerodynamics, sensor lag and MCS/CAS	Aerodynamics alone
0.0000 + 0.0000i	-0.0042 + 0.0399i
0.0178 + 0.0259i	-0.0042 - 0.0399i
0.0178 - 0.0259i	-0.6622 + 1.1988i
-0.1000 + 0.0000i	-0.6622 - 1.1988i
-0.1000 + 0.0000i	
-0.3298 + 0.0000i	
-1.2223 + 1.2490i	
-1.2223 - 1.2490i	
-5.0000 + 0.0000i	
-19.8335 + 0.0000i	

Table 5-3. Eigenvalues for ZAPO Linear Model

5.2.1 Nine State Aircraft Model

Since the thrust is stabilized in the zoom and pushover model, the engine lag state, and thrust control will be dropped in the subsequent analyses.

Frequency responses of the five outputs to sinusoidal elevator inputs are shown in Figures 5-1 and 5-2.

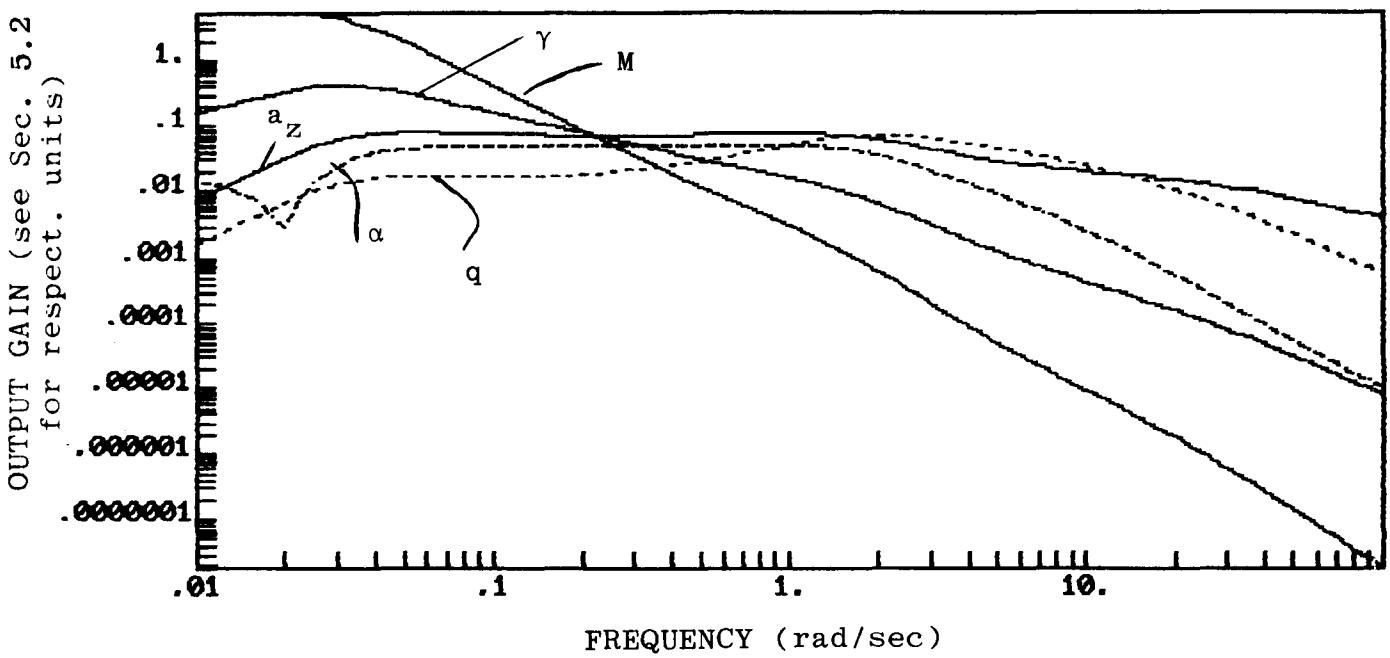

Figure 5-1. Output Magnitude Responses to Elevator -- Nine State Aerodynamic and MCS/CAS Model

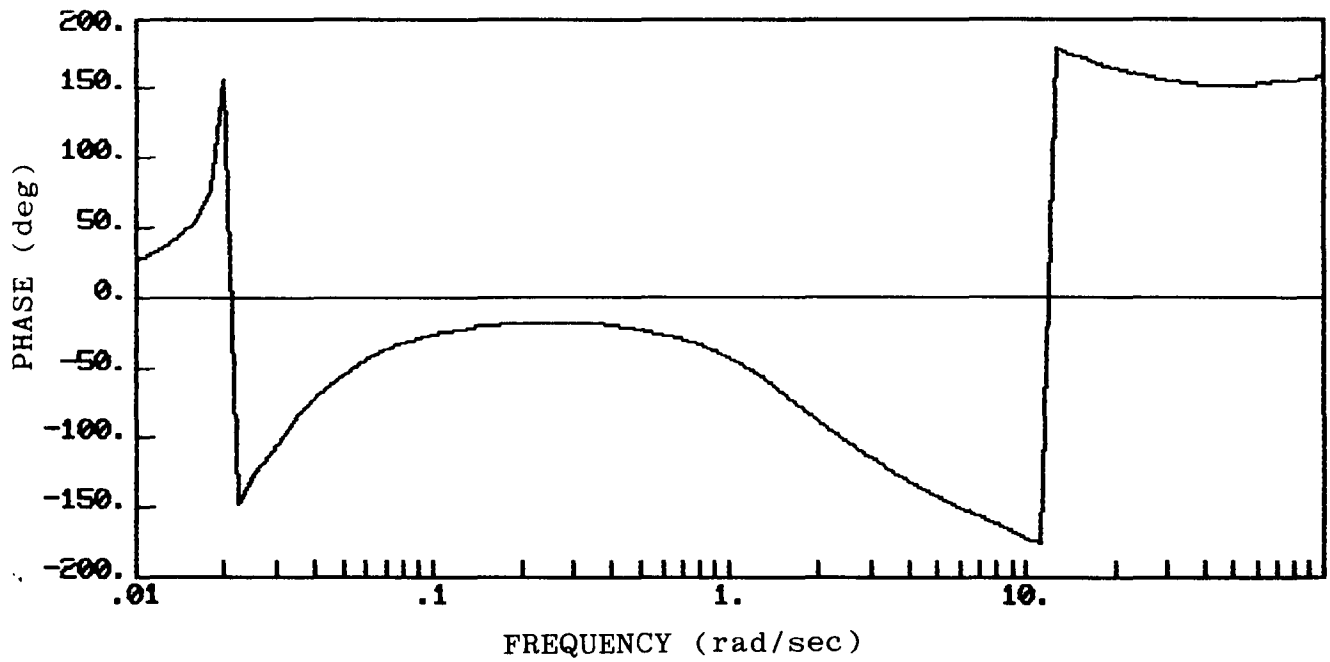

Figure 5-2. Phase of $\alpha(s)/\delta_e(s)$ -- Nine State Aircraft and MCS/CAS

The $\alpha(s)/\delta_e(s)$ transfer function has the dip at low frequency because of zeroes produced by the MCS/CAS which attempt to mitigate the phugoid mode (see Figures 5-3 and 5-4).

The high phase response is caused by the unstable phugoid mode.

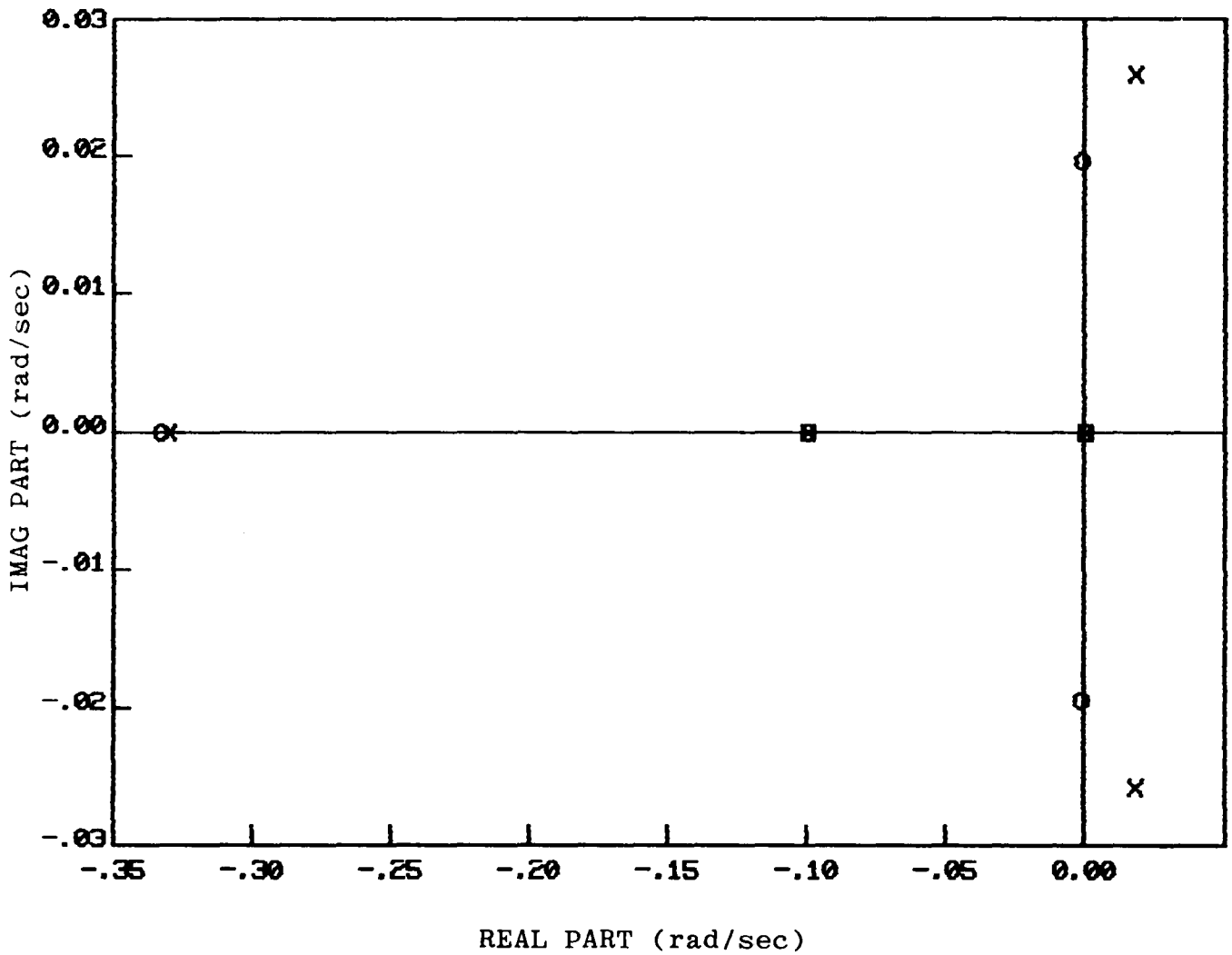

Figure 5-3. $\alpha(s)/\delta_e(s)$ -- Low Frequency Poles and Zeros -- Nine State Aircraft and MCS/CAS

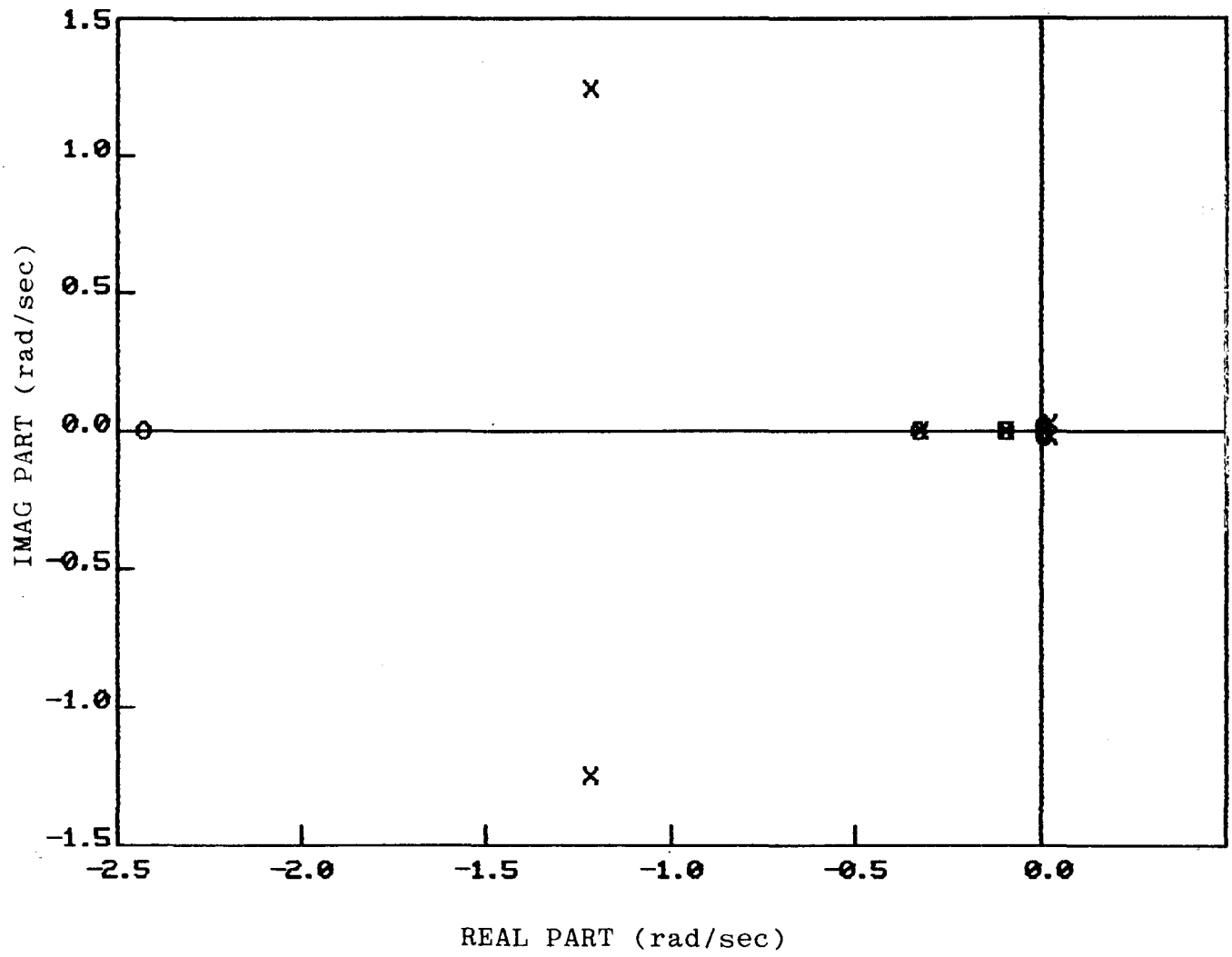

Figure 5-4. $\alpha(s)/\delta_e(s)$ -- Mid-Frequency Poles/Zeros -- Nine State Aircraft and MCS/CAS

The residues of these transfer functions are given below in Table 5-4.

Eigenvalues	a_z/δ_e	α/δ_e	q/δ_e	γ/δ_e	M/δ_e
0.0000 + 0.0000i	0.0000	0.0000	0.0000	0.0000	0.0000
0.0178 + 0.0259i	-0.0026	0.0018	0.0006	0.0150	0.0183
0.0178 - 0.0259i	0.0009	0.0002	-0.0002	0.0110	-0.1753
-0.1000 + 0.0000i	0.0000	0.0000	0.0000	0.0000	-0.0163
-0.3298 + 0.0000i	-0.0003	0.0002	0.0000	-0.0002	-0.0001
-1.2223 + 1.2490i	-0.0800	0.0441	0.0879	-0.0203	-0.0020
-1.2223 - 1.2490i	-0.1557	0.0999	-0.1441	-0.0063	0.0010
-5.0000 + 0.0000i	0.0104	-0.0607	0.3012	0.0004	0.0000
-19.8335 + 0.0000i	0.4704	0.0146	-0.3897	0.0050	0.0001

Table 5-4 Residues of Elevator Transfer Functions (Nine State Aerodynamics and MCS/CAS)

These residues indicate that a good model can be achieved with five states, by including the phugoid and short period modes and the control system lag at 5 radians/sec. Such a model would be useful for flight test trajectory control signals fed directly into the aircraft control system through the autopilot. This reduced order model is detailed in the next subsection.

5.2.2 Five State Aircraft Reduced-Order Model

The nine state model of the previous subsection, reduced to match at zero frequency by the method of Section 4.1 gives the following equations

$$\dot{x}_r = F_r x_r + G_r u + {}_r w \ ,$$

$$y = H_{y_r} x_r + J_{u_r} u \ ,$$

$$z = H_{z_r} x_r + J_{w_r} w \ ,$$

with the matrices given by

$$F_r = \begin{bmatrix} 0.0178 & 0.0259 & 0.0000 & 0.0000 & 0.0000 \\ -0.0259 & 0.0178 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -1.2223 & 1.2490 & 0.0000 \\ 0.0000 & 0.0000 & -1.2490 & -1.2223 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -5.0000 \end{bmatrix}$$

$$G_r \begin{bmatrix} 0.2905 \\ -0.0541 \\ -0.1374 \\ 0.0904 \\ 12.1783 \end{bmatrix}$$

$$\Gamma_r = \begin{bmatrix} 0.5576 \\ 0.1872 \\ 2.0404 \\ -0.7345 \\ 0.0000 \end{bmatrix}$$

$$H_{y_r} = \begin{bmatrix} -0.0092 & -0.0014 & -0.1140 & -1.0581 & 0.0009 \\ 0.0060 & -0.0017 & 0.1101 & 0.6546 & -0.0050 \\ 0.0021 & 0.0004 & -0.9280 & -0.4379 & 0.0247 \\ 0.0431 & -0.0459 & 0.0822 & -0.0998 & 0.0000 \\ 0.1693 & 0.5720 & 0.0133 & -0.0014 & 0.0000 \end{bmatrix}$$

$$J_{u_r} = \begin{bmatrix} -0.0696 \\ 0.0012 \\ -0.0197 \\ -0.0002 \\ -0.1631 \end{bmatrix}$$

$$H_{z_r} = \begin{bmatrix} -0.0092 & -0.0014 & -0.1140 & -1.0581 & 0.0009 \\ 0.0060 & -0.0017 & 0.1101 & 0.6546 & -0.0050 \\ 0.0021 & 0.0004 & -0.9280 & -0.4379 & 0.0247 \\ 0.0371 & -0.0442 & -0.0279 & -0.7544 & 0.0050 \\ 0.1693 & 0.5720 & 0.0133 & -0.0014 & 0.0000 \end{bmatrix}$$

$$J_{w_r} = \begin{bmatrix} 0.4187 \\ -1.8776 \\ 0.0034 \\ 2.1376 \\ -2.5170 \end{bmatrix}$$

The frequency response of this five state model is shown in Figure 5-5. The model is only good up to about 4-12 Radians/sec and only up to .1 radians in the mach response which is primarily due to its own lag at .1 second which was not included in the model.

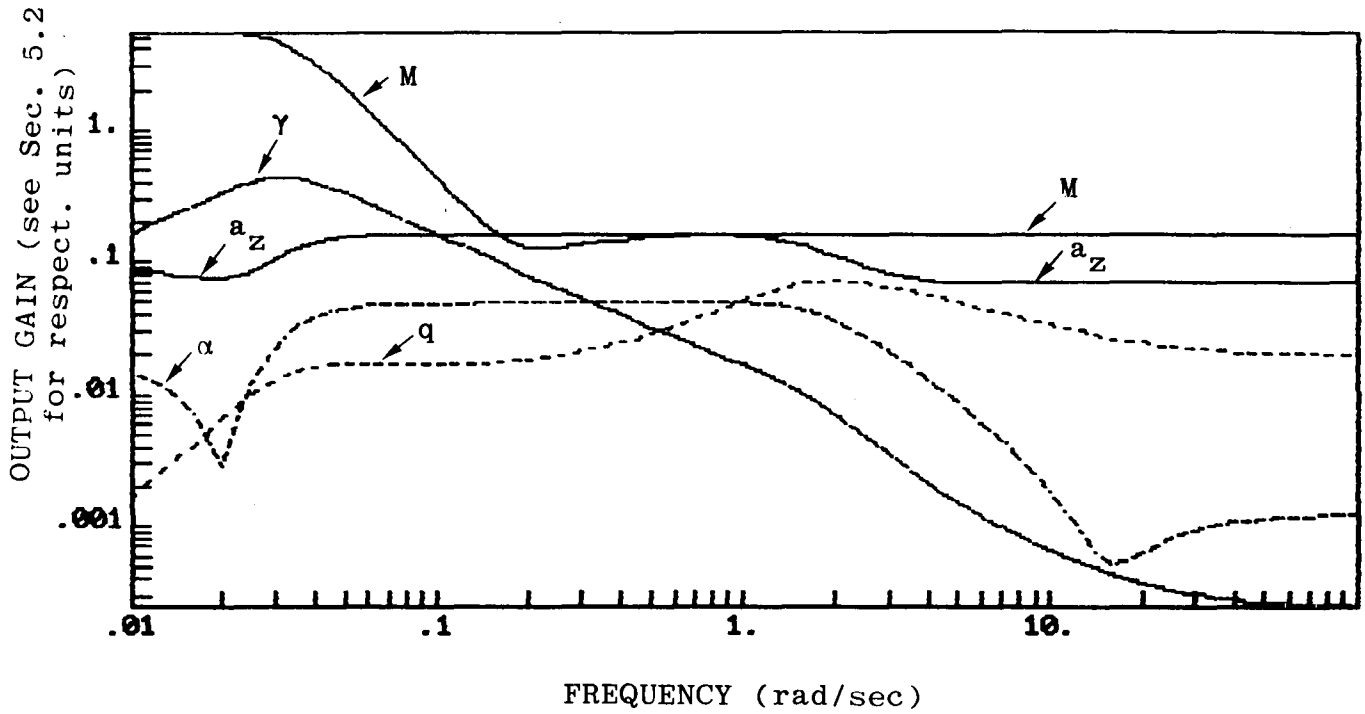

Figure 5-5 Transfer Function Magnitudes - Five State Aircraft and MCS/CAS (Reduced Order Model Matched at Zero Frequency)

5.2.3 Thirteen State Piloted Aircraft Model

The four state pilot model of Section 3.5 is for a $1/s^2$ plant. The pilot adapts his own gain to make the plant look like the dynamics he wants. At this point we can ignore the gain question of the pilot until after the compensator has been designed, then adjust the compensator gain to simulate the pilot's action.

The unreduced design model is given by

$$\begin{bmatrix} \dot{x}_{A/C} \\ \dot{x}_p \end{bmatrix} = \begin{bmatrix} F_{A/C} & G_{A/C}H_p \\ 0 & F_p \end{bmatrix} \begin{bmatrix} x_{A/C} \\ x_p \end{bmatrix} + \begin{bmatrix} 0 \\ G_p \end{bmatrix} + \begin{bmatrix} \Gamma & 0 \\ 0 & \Gamma_p \end{bmatrix} \begin{bmatrix} \alpha_{gust} \\ w_p \end{bmatrix} ,$$

$$y = [H_{y_{A/C}} \quad 0] ,$$

$$z = [H_{z_{A/C}} \quad 0] + v .$$

The magnitude and phase responses of the five outputs to displayed elevator inputs are given by Figure 5-6 and 5-7. These frequency responses are not surprising when looking at transfer function of the linear pilot model (see Figures 5-8 and 5-9).

The response of this thirteenth order system to gust inputs is shown in Figure 5-10.

The residues of the elevator transfer functions are the basis for model reduction. They are shown in Table 5-5.

The pilot model dominates, so these 4 modes together with the unstable phugoid mode comprise the reduced order design model described in the next subsection.

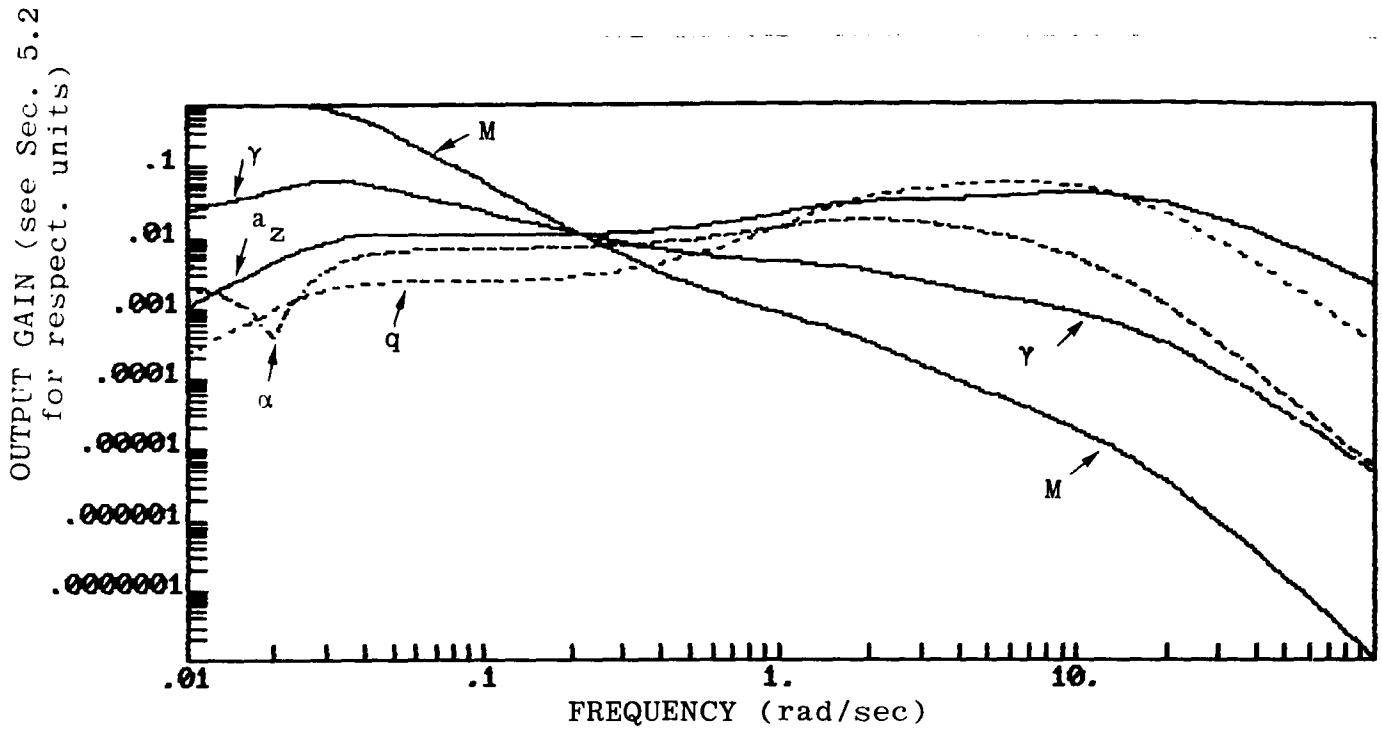

Figure 5-6 Response to Displayed Elevator Command - 13 State A/C and Pilot Model

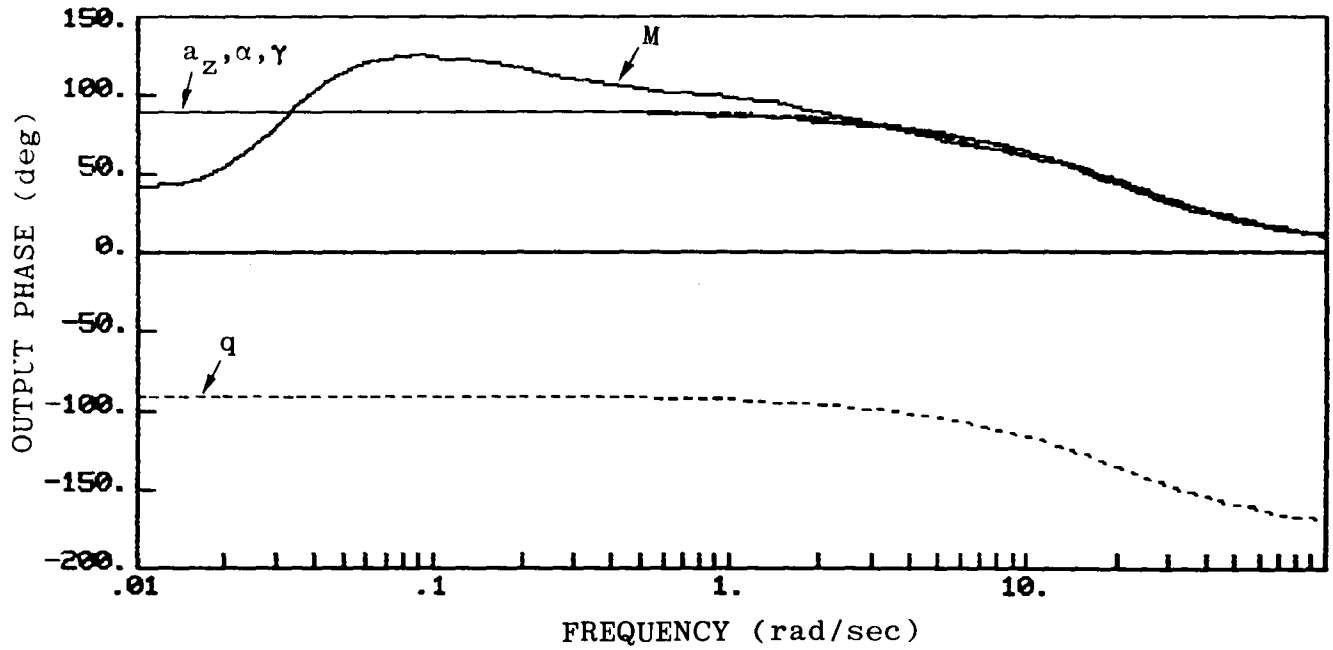

Figure 5-7 Phase Response to Displayed Elevator Command - 13 State A/C and Pilot Model

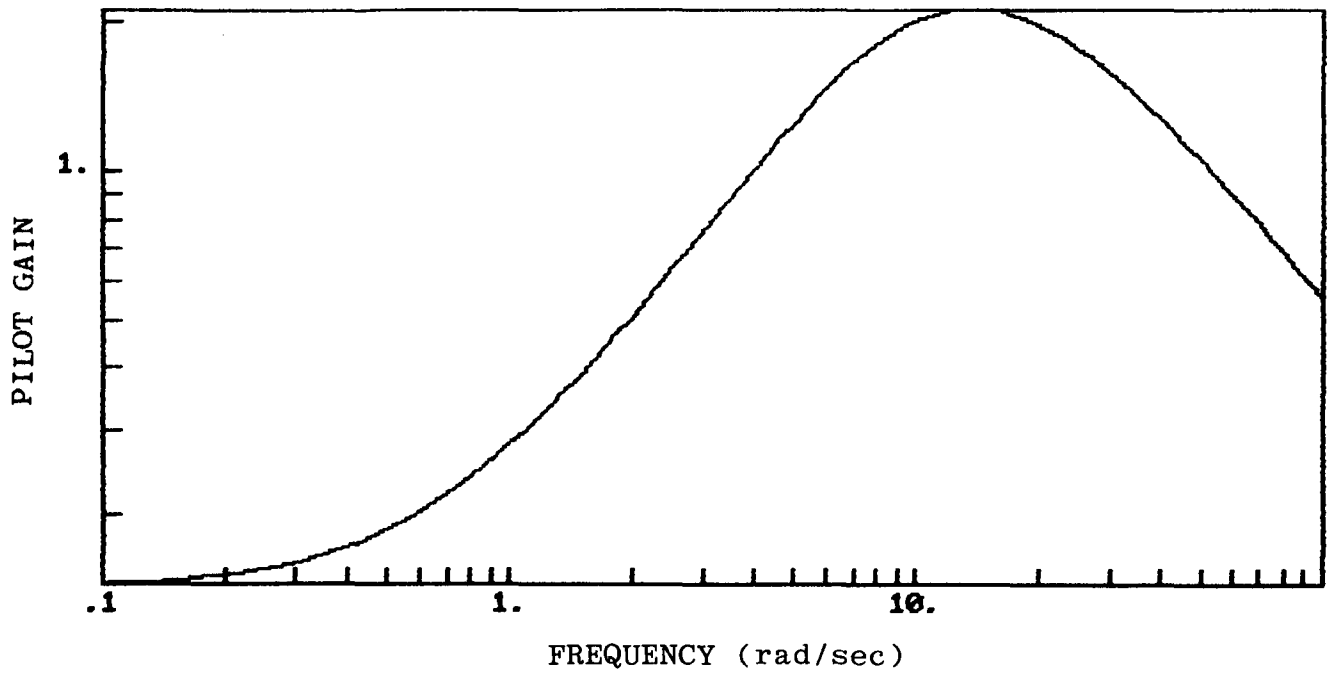

Figure 5-8 Pilot Model Magnitude

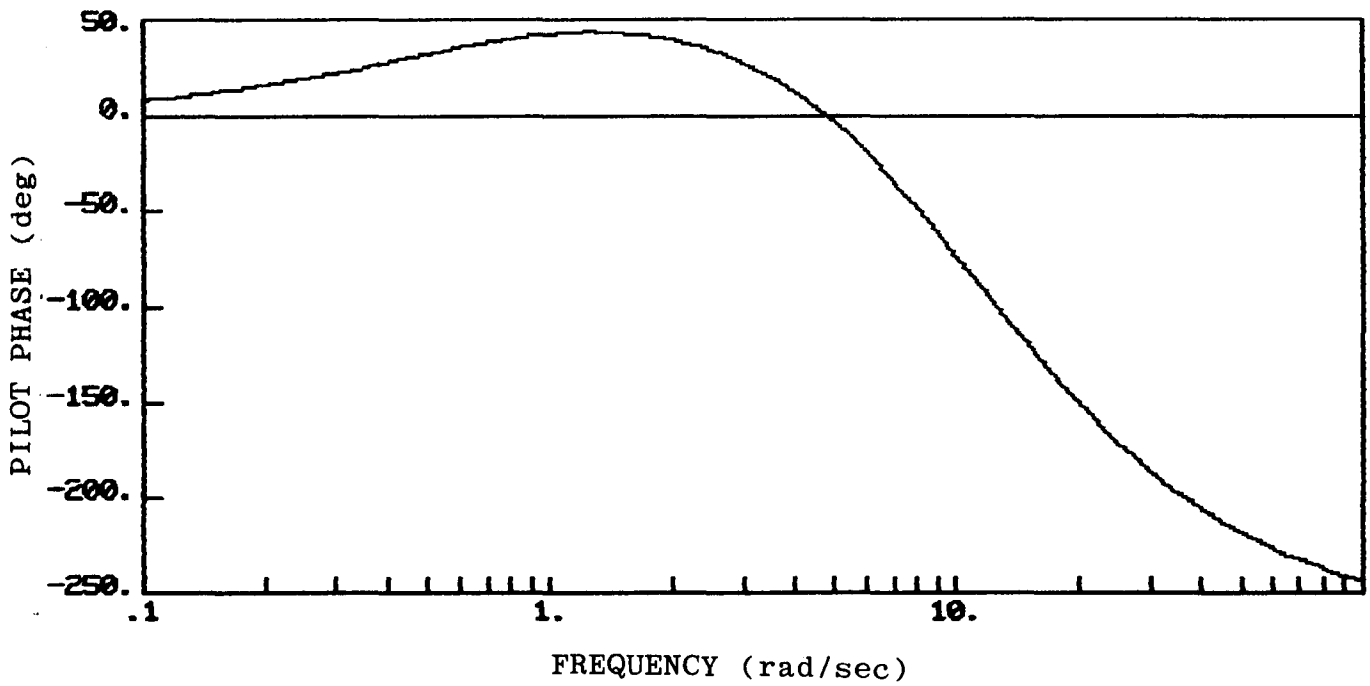

Figure 5-9 Pilot Model Phase Response

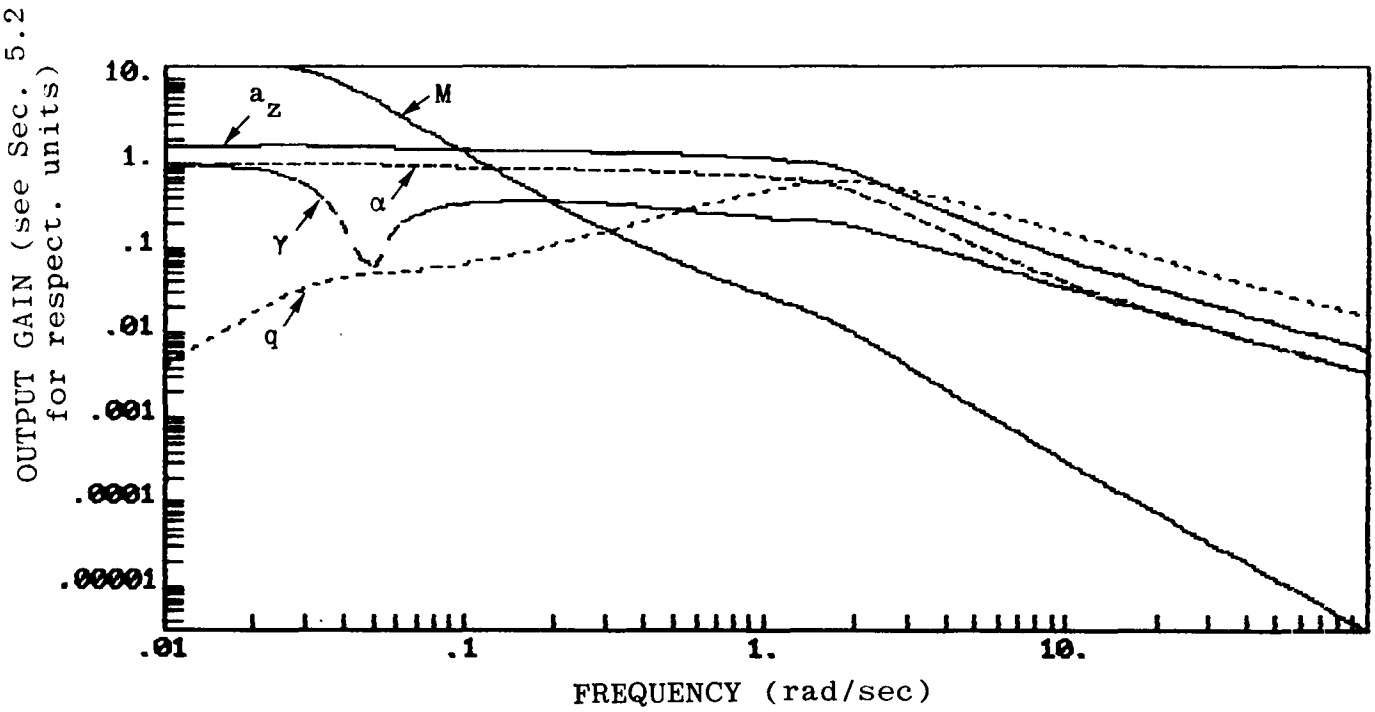

Figure 5-10 Response to Gust - 13 State Piloted Aircraft Model

Table 5-5 Elevator Residues - Thirteen State Piloted Aircraft

Eigenvalues	a_z/δ_e	α/δ_e	q/δ_e	γ/δ_e	M/δ_e
0.0000 + 0.0000i	-3.7851D-32	-3.1477D-17	-8.6754D-31	-1.1798D-17	8.4780D-15
0.0178 + 0.0259i	6.7004D-06	-4.8157D-06	-1.5159D-06	-4.0116D-05	-3.1957D-05
0.0178 - 0.0259i	-2.6021D-06	-2.7827D-07	6.7240D-07	-2.7340D-05	4.5871D-04
-0.1000 + 0.0000i	-3.8206D-21	2.5099D-21	2.0833D-21	2.7531D-20	3.5578D-05
-0.3298 + 0.0000i	3.2125D-07	-2.1251D-07	1.8614D-09	2.0687D-07	7.6295D-08
-1.2223 + 1.2490i	1.2471D-03	-8.0742D-04	1.3152D-03	3.3311D-05	-1.0549D-05
-1.2223 - 1.2490i	-8.3688D-04	4.7492D-04	6.0570D-04	-1.7944D-04	-1.5658D-05
-5.0000 + 0.0000i	-2.7288D-04	1.5872D-03	-7.8779D-03	-1.1590D-05	4.1504D-07
-8.0000 + 0.0000i	-1.0129D+00	-4.0442D-01	3.4505D+00	-2.6886D-02	-8.5858D-04
-9.5240 + 0.0000i	1.2567D+01	2.7990D+00	-2.9326D+01	2.8021D-01	7.1613D-03
-10.0000 + 0.0000i	-1.3282D+01	-2.5525D+00	2.8345D+01	-2.8205D-01	-6.8069D-03
-14.0000 + 0.0000i	2.2990D+00	1.7493D-01	-2.9372D+00	3.4872D-02	5.8182D-04
-19.8335 + 0.0000i	-5.7243D-01	-1.7783D-02	4.7425D-01	-6.1291D-03	-7.1287D-05

5.2.4 Six State Piloted Aircraft Reduced-Order Model

The reduced piloted aircraft model is of the same form as the equations of Section 5.2.2, with the following matrices

$$F = \begin{bmatrix} 0.0178 & 0.0259 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ -0.0259 & 0.0178 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & -8.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & -9.5240 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -10.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -14.0000 \end{bmatrix}$$

$$G = 1.0D+04 * \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0985 \\ -4.2976 \\ -0.1120 \\ 0.3172 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} 0.5576 & -16.2773 \\ 0.1872 & 2.9859 \\ 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \\ 0.0000 & 44.2543 \\ 0.0000 & 0.0000 \end{bmatrix}$$

$$H_y = \begin{bmatrix} -0.0092 & -0.0014 & 0.0577 & 0.0164 & -0.6655 & -0.0407 \\ 0.0060 & -0.0017 & 0.0230 & 0.0037 & -0.1279 & -0.0031 \\ 0.0021 & 0.0004 & -0.1965 & -0.0383 & 1.4202 & 0.0520 \\ 0.0431 & -0.0459 & 0.0015 & 0.0004 & -0.0141 & -0.0006 \\ 0.1693 & 0.5720 & 0.0000 & 0.0000 & -0.0003 & 0.0000 \end{bmatrix}$$

$$J_u = \begin{bmatrix} 1.6135 \\ 0.0398 \\ -1.2966 \\ 0.0208 \\ -0.0192 \end{bmatrix}$$

$$H_z = \begin{bmatrix} -0.0092 & -0.0014 & 0.0577 & 0.0164 & -0.6655 & -0.0407 \\ 0.0060 & -0.0017 & 0.0230 & 0.0037 & -0.1279 & -0.0031 \\ 0.0021 & 0.0004 & -0.1965 & -0.0383 & 1.4202 & 0.0520 \\ 0.0491 & -0.0476 & 0.0246 & 0.0040 & -0.1420 & -0.0037 \\ 0.1693 & 0.5720 & 0.0000 & 0.0000 & -0.0003 & 0.0000 \end{bmatrix}$$

$$J_w = \begin{bmatrix} 1.5377 & 6.8958 \\ -0.9490 & -2.0947 \\ 0.0185 & -7.2272 \\ -0.5339 & -1.3389 \\ -2.5086 & 9.2763 \end{bmatrix}$$

The frequency magnitude response to elevator inputs for this sixth order reduced order model is given in Figure 5-11. This model is considerably different from the linear evaluation (thirteenth order) model above one radian per second. However, the model information at low frequency is quite good and, with frequency shaped control methods, can be used to develop a robust flight test trajectory controller. Reducing the thirteenth order model at a higher frequency does not significantly improve the flight path angle response which can be observed in Figures 5-12 and 5-13. Figures 5-14 and 5-15 show the response of this reduced order design model to gusts and pilot motor noise.

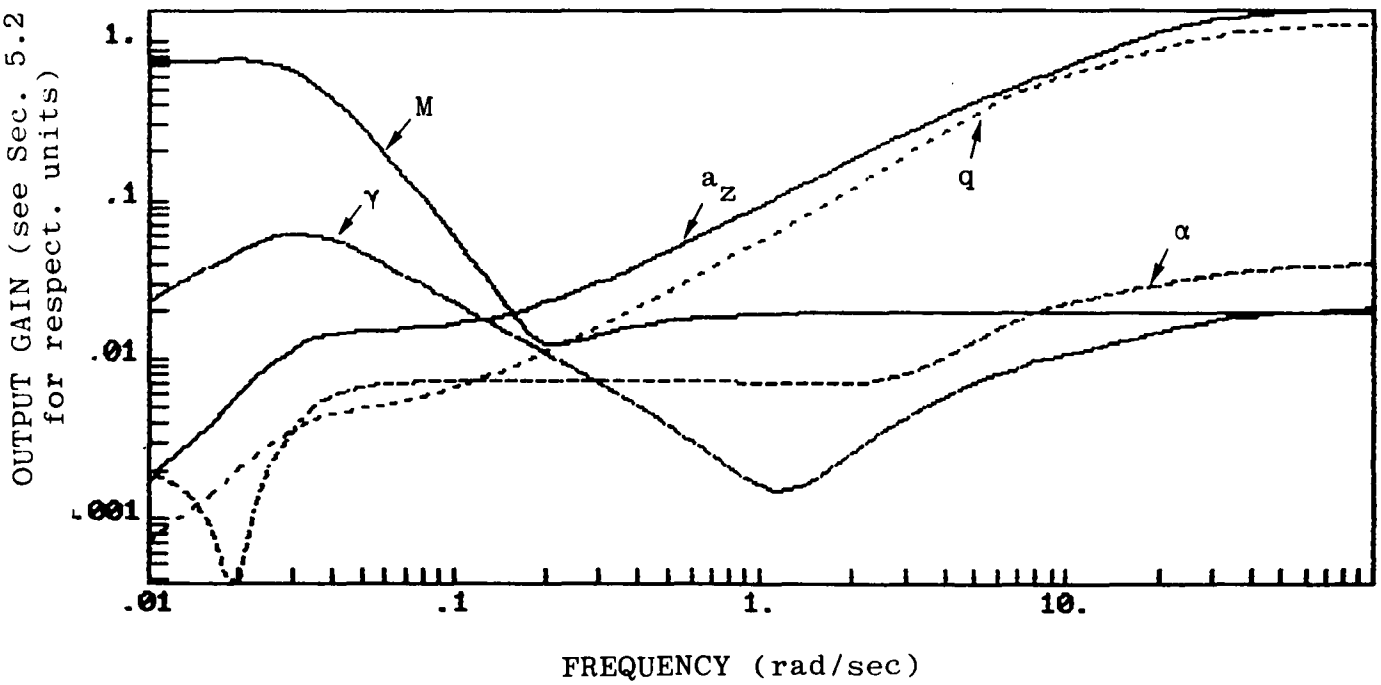

Figure 5-11 Magnitude Response to Displayed Elevator Command - Six State Reduced Piloted A/C Model (Reduced at Zero Frequency)

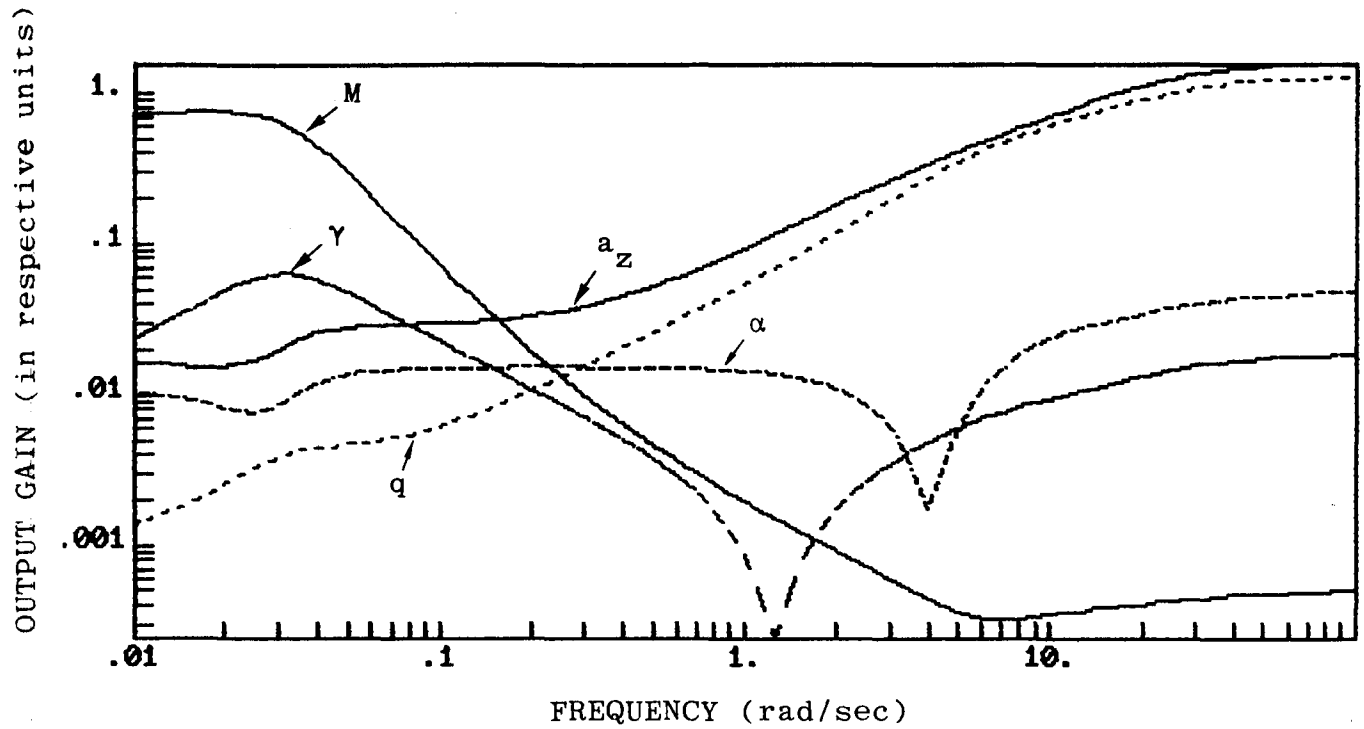

Figure 5-12 Response to Displayed Elevator - Six State Piloted A/C Model (Reduced at 1 Rad/Sec)

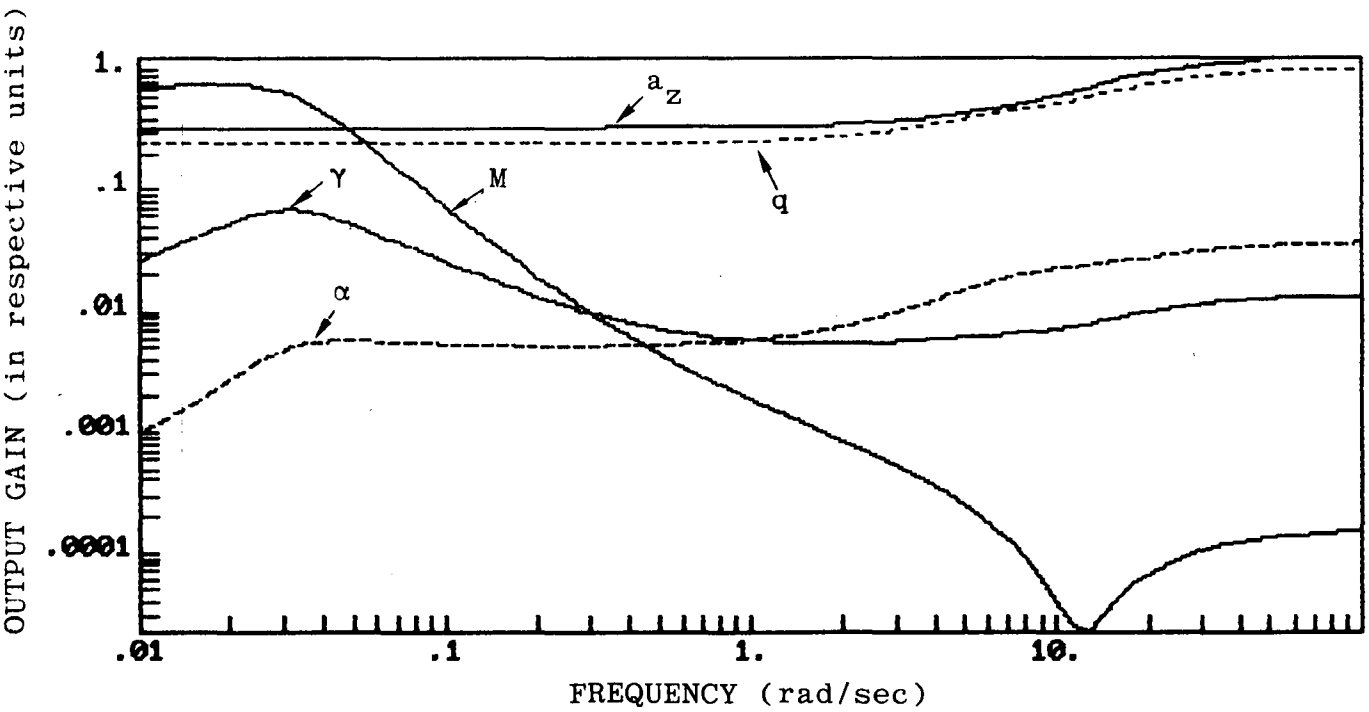

Figure 5-13 Response to Displayed Elevator - Six State Piloted A/C Model (Reduced at 10 Rad/Sec)

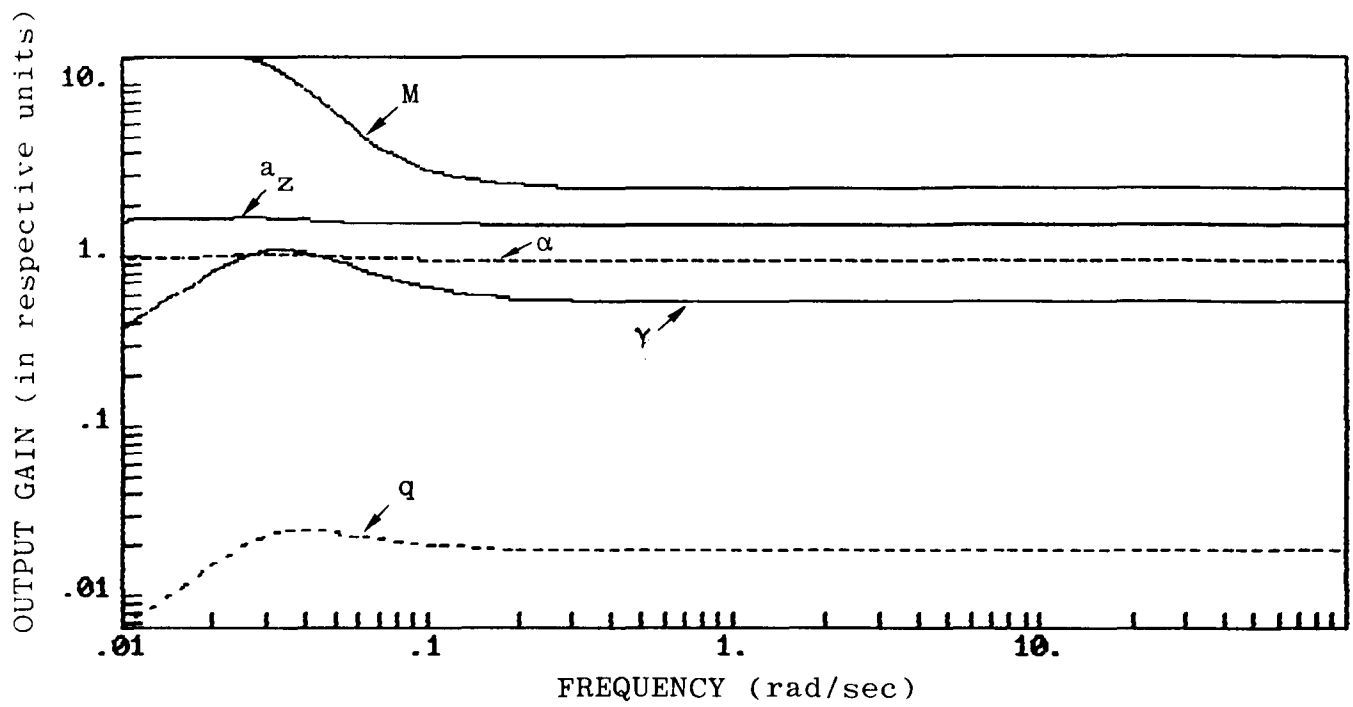

Figure 5-14 Response to Gust - Six State Piloted A/C Model (Reduced at Zero Frequency)

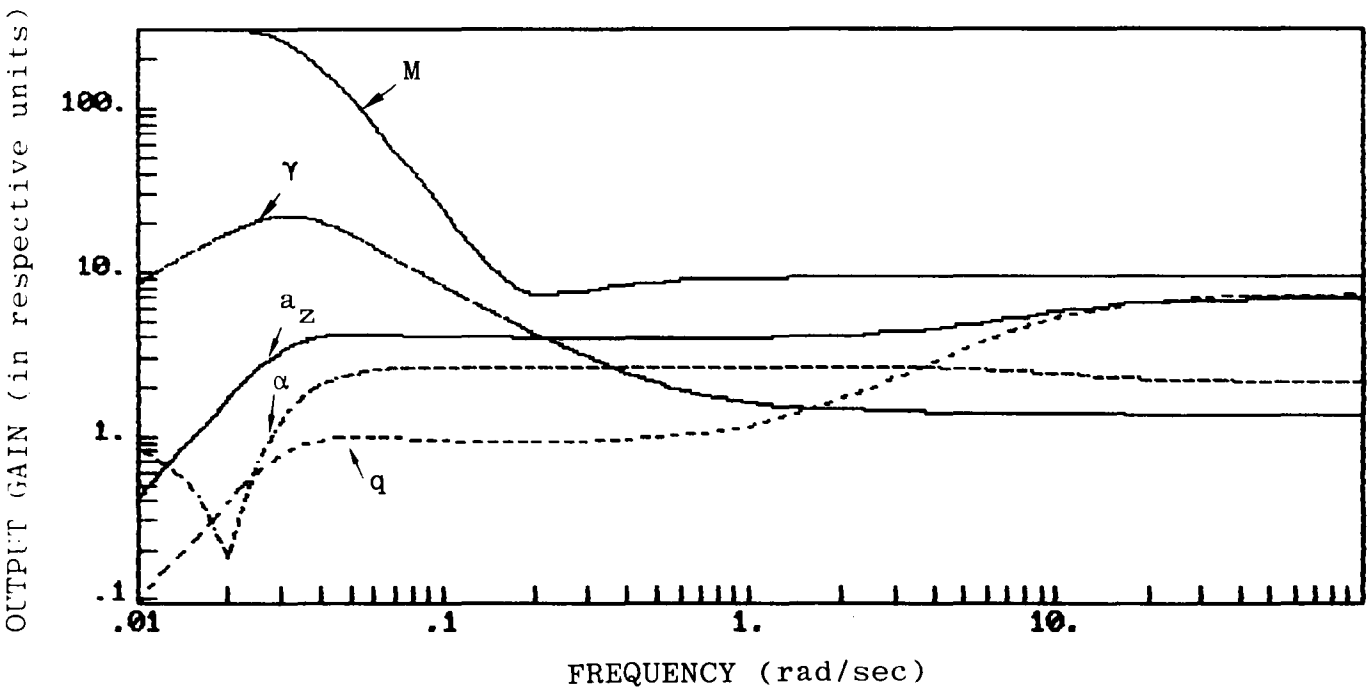

Figure 5-15 Response to Pilot Motor Noise - Six State Piloted A/C Model (Reduced at Zero Frequency)

5.3 FLIGHT TEST TRAJECTORY CONTROL (FTTC) COMPENSATOR

Using the reduced order design model of the piloted aircraft from the previous section, a standard linear quadratic regulator could be designed with the following quadratic costs on outputs and inputs,

$$A_y = \text{diag}\ [.01,\ .01,\ .01,\ 10,\ 1]\ ,\ B_{\delta e} = 1\ .$$

This yields the following control gains and closed loop regulator eigenvalues shown below,

$$C = [1.4068 \quad -1.7544 \quad .0011 \quad .0003 \quad -.0106 \quad -.0006]\ ,$$

with $\lambda_i\ (F_r - G_r C)$

$$-12.511 \pm 1.8195i$$

$$-8.0244 \pm .9075i$$

$$-.0210 \pm .0280i\ .$$

The closed loop regulator response to incremental elevator is shown in Figure 5-16.

The poor knowledge of the plant at high frequency as shown in Figure 5-16 cannot be overcome with regulator design. The next subsection shows a frequency shaped controller which uses the high fidelity of the model at low frequency, gives acceptable performance at mid-frequency where the pilot works, and rolls off the controller at high frequency.

5.3.1 Regulator Design

A more robust controller with respect to the reduced order modeling errors can be obtained by using frequency shaped control concepts as described in Appendix A. The following performance index

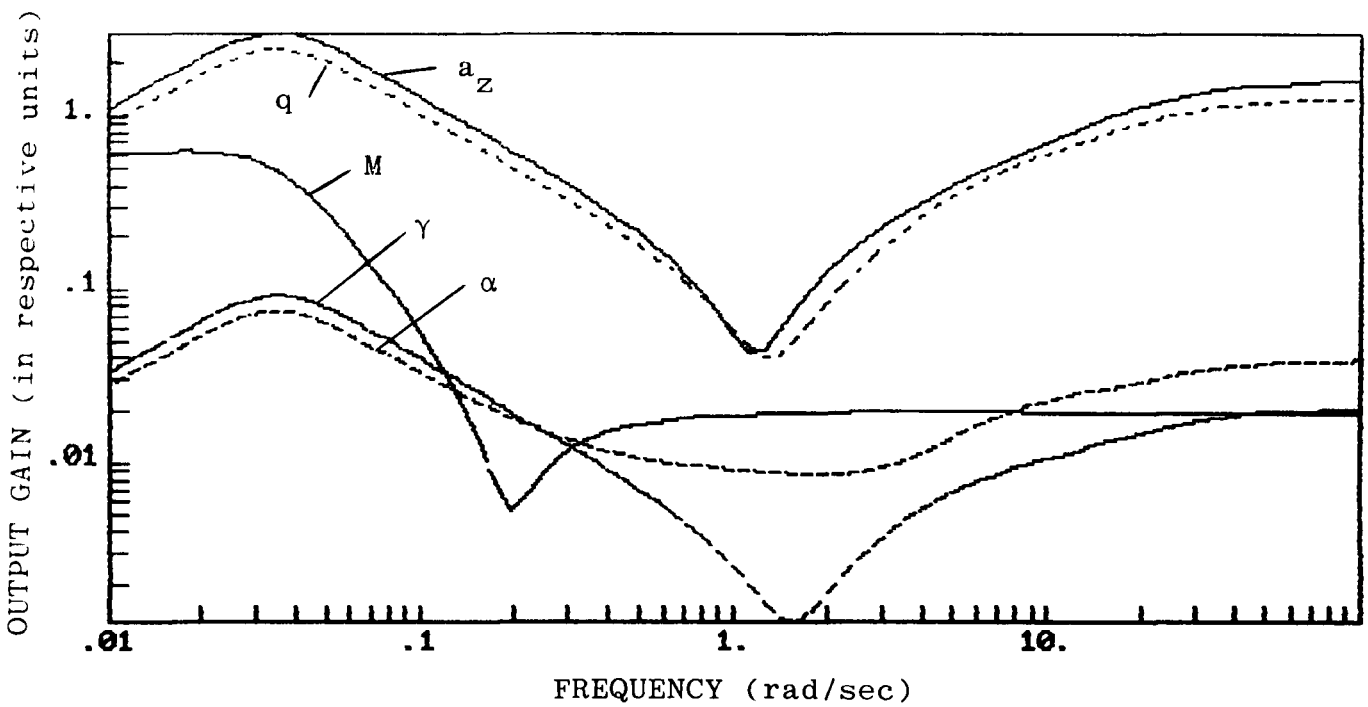

Figure 5-16. Response to Elevator with Full State Feedback -- Six State Piloted A/C
(Plant reduced at zero frequency)

$$\lim_{w\to\infty} \int_{-w}^{w} \left[\frac{A_t}{(1+w^2)^2} + (.01+w^2)\, B_{\delta e} \right] dw \ ,$$

can be implemented by the following equations

$$\frac{d}{dt} \begin{bmatrix} x \\ z_1 \\ u \end{bmatrix} = \begin{bmatrix} F & 0 & 0 \\ GH_{a_z} & F_1 & G_1 J_{u_{a_z}} \\ 0 & 0 & F_2 \end{bmatrix} dx + \begin{bmatrix} 0 \\ 0 \\ G_2 \end{bmatrix} \bar{u} \ ,$$

where $F_1 = \begin{bmatrix} -1 & 1 \\ & -1 \end{bmatrix}$, $G_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $F_2 = -.1$, $G_2 = .1$,

and F, G, H_{a_z} , and Ju_{a_z} are given in Subsection 5.2.4,

with the performance index as

$$\lim_{T\to\infty} \frac{1}{2T} \int_{0}^{T} \left(z_{1_1}^2 + \bar{u}^2 \right) dt \ .$$

The control law is given by

$$\bar{u} = -C_1 x - C_2 z_1 - C_3 u \ ,$$

with feedback gains and closed loop regulator eigenvalues as follows:

$$C_A = [C_1, C_2, C_3] =$$

```
 COLUMNS  1 THRU  8
 14.2516  -14.5265   0.0000   0.0000   0.0000   0.0000   0.0009   0.0003

 COLUMNS  9 THRU  9
  0.7124
```

$$\lambda_i(F_A - G_A C_A) = \begin{matrix} -14.0000 - 0.0000i \\ -9.5240 - 0.0000i \\ -10.0000 - 0.0000i \\ -8.0000 - 0.0000i \\ -1.0000 + 0.0032i \\ -1.0000 - 0.0032i \\ -0.0671 + 0.2529i \\ -0.0671 - 0.2529i \\ -0.0014 + 0.0000i \end{matrix}$$

The control law for u can be obtained from the ninth state of the compensator as shown in Subsection 5.3.3. Our control $u = \delta_e$ is primarily used to control flight path angle. By rolling off the acceleration, the derivative of flight path angle, at high frequency, the controller achieves good roll off in all of the outputs as shown in Figure 5-17.

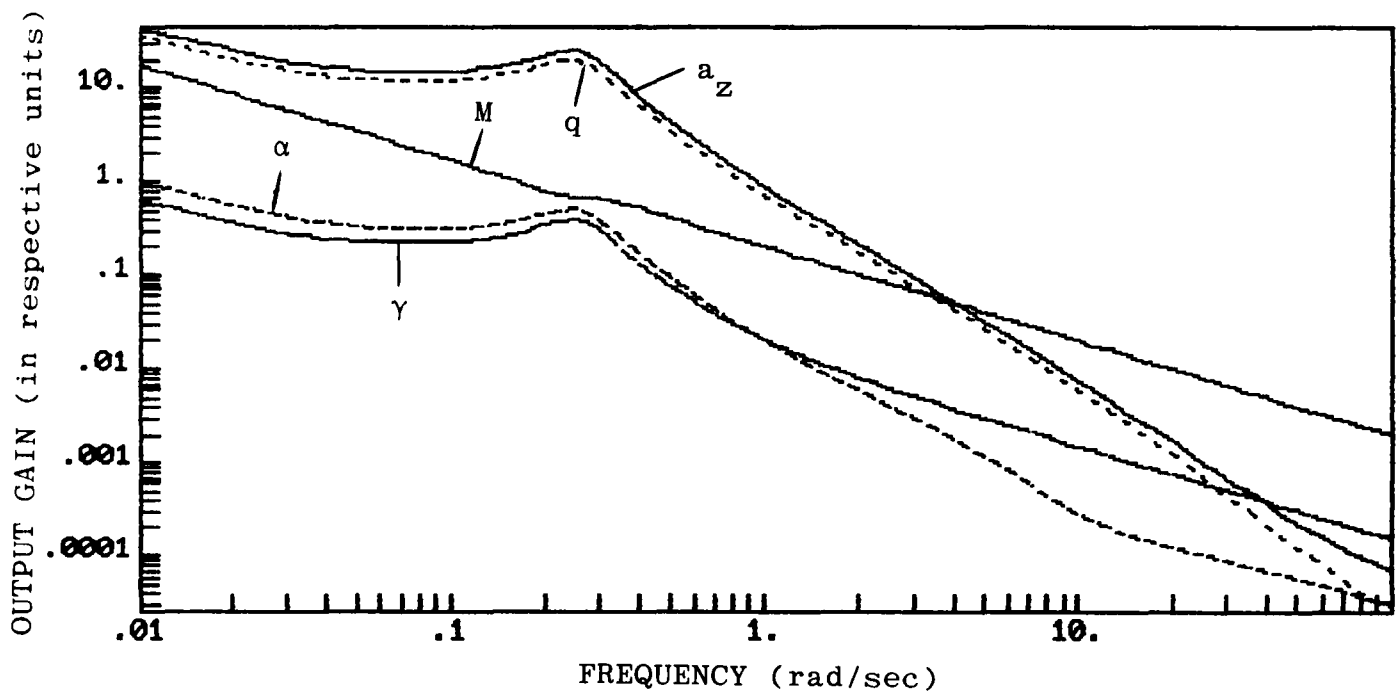

Figure 5-17. Closed Loop Regulator Response to Gust (Six piloted A/C states with three shaping states)

5.3.2 Estimator Design

An estimator can be designed for the six state piloted aircraft model with the gust and measurement power spectral densities given in Subsection 5.2.3. Using all five measurements and $w(t) = \alpha_{gust}$ input the performance index is

$$J = E\left[w^T(t)Q^{-1}w(t) + v^T(t)R^{-1}v(t)\right] ,$$

yielding an estimator feedback gain matrix K to implement the filter equations

$$\dot{\hat{x}} = F\hat{x} + Gu + K(z - H_z x - J_u u) .$$

The gains and filter closed loop eigenvalues are shown below.

K =

0.1409	-0.2728	0.2119	0.5487	-0.0320
0.0486	-0.1244	-0.0345	-0.3664	0.1011
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000

$\lambda_i(F - KH_z)$ =

-0.0294 + 0.0295i
-0.0294 - 0.0295i
-14.0000 + 0.0000i
-8.0000 + 0.0000i
-9.5240 + 0.0000i
-10.0000 + 0.0000i

5.3.3 Reduced Order Compensator

The full state compensator is given by

$$\frac{d}{dt}\begin{bmatrix} x \\ z_1 \\ u \end{bmatrix} = \begin{bmatrix} F-kH_z & O & G-kJu_zH_c \\ GH_{a_z} & F_1 & G_1Ju_{a_z} \\ -G_2C_1 & -G_2C_2 & F_2-G_2C_3 \end{bmatrix}\begin{bmatrix} x \\ z_1 \\ u \end{bmatrix} + \begin{bmatrix} k \\ O \\ O \end{bmatrix} z$$

with $H_c = [0\ 0\ 1]$.

The feedforward of the control into the measurements requires a negative feedback in the compensator. The residues of this transfer function are given below in Table 5-6

Eigenvalues	U^*_{D/a_z}	$U_{D/\alpha}$	$U_{D/q}$	$U_{D/\gamma}$	$U_{D/M}$
-0.0793 + 0.0000i	-0.2075	0.5248	0.1258	1.4533	-0.4092
-0.0807 + 0.2289i	0.2074	-0.5247	-0.1259	-1.4536	0.4093
-0.0807 - 0.2289i	-0.5675	0.9060	-1.5388	-5.7511	0.8434
-1.0000 + 0.0033i	0.0000	0.0000	0.0000	0.0000	0.0000
-1.0000 - 0.0033i	0.0005	-0.0007	0.0014	0.0053	-0.0008
-8.3719 + 0.6590i	-0.0005	0.0008	-0.0013	-0.0049	0.0007
-8.3719 - 0.6590i	0.0019	-0.0030	0.0052	0.0193	-0.0028
-10.9263 + 0.0000i	0.0006	-0.0010	0.0017	0.0065	-0.0010
-13.8433 + 0.0000i	-0.0001	0.0002	-0.0003	-0.0012	0.0002

Table 5-6. Full State Compensator Residues.

*U_D is displayed control (elevator) command

This compensator can be well approximated with a three state compensator using modal reduction just as the original model was reduced (see Section 4.1). The reduced order compensator is given by

$$\dot{x}_c = F_c x_c + G_c z \ , \quad u = H_c x_c \ ,$$

with

FC =

-1.9338D-01	2.5536D-01	1.3422D-01
-2.3696D-01	2.0763D-01	6.4751D-02
-5.5726D-01	3.0108D-01	-2.5491D-01

GC =

-8.9338D-01	1.5645D+00	-1.9311D+00	-6.5153D+00	8.1542D-01
-1.1924D+00	2.2681D+00	-1.9376D+00	-5.3908D+00	4.2135D-01
-5.9027D-01	8.0340D-01	-2.0951D+00	-8.5362D+00	1.3927D+00

HC =

4.4421D-01	-2.4000D-01	-1.8743D-01 .

The magnitude and frequency responses of these five transfer functions are shown in Figures 5-18 and 5-19. On the same scales the ninth order frequency response plots are indistinguishable as would be expected from the size of the residues.

5.3.4 Controlled Plant Dynamics

The compensator can be further analyzed by investigating the effect of the dynamic compensator on the full thirteen state aircraft and MCS/CAS model, G(s). Viewing the pilot as a compensator, P(s), for this system, we are interested in the transfer function from the aircraft stick to the displayed elevator command to which he responds (see Figure 5-20 below).

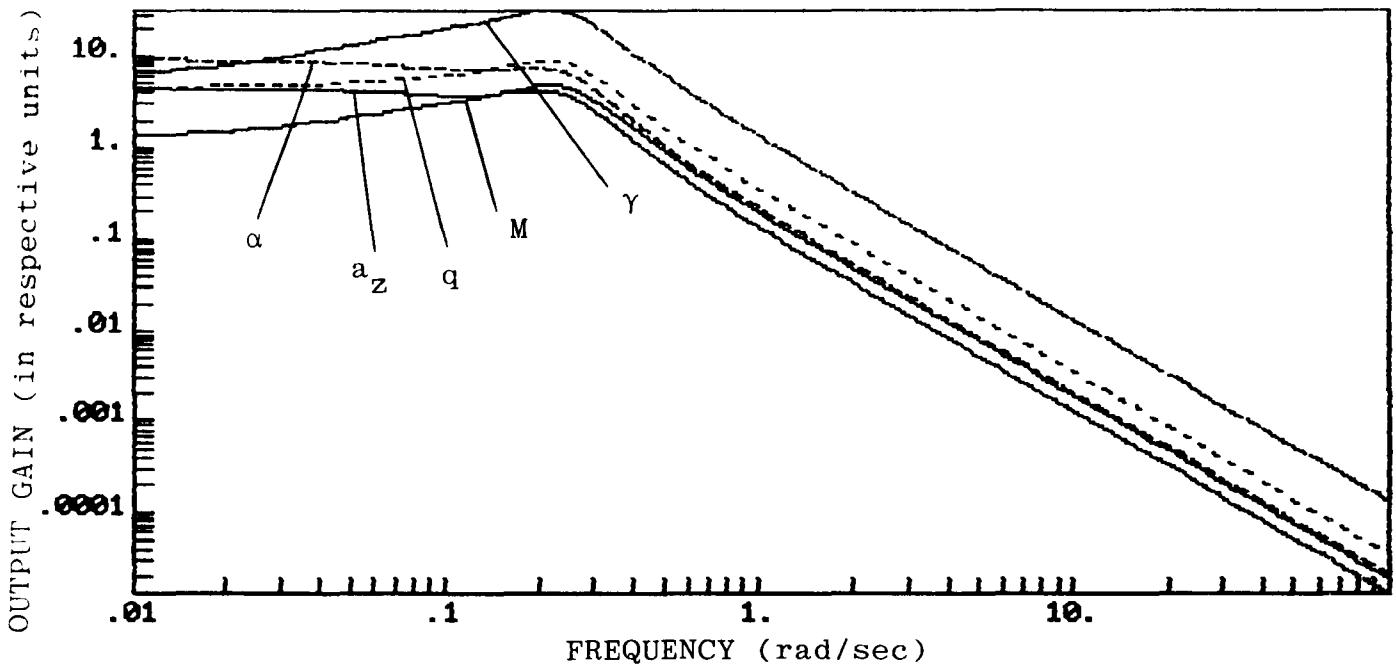

Figure 5-18. Three State Reduced Order Compensator Magnitude Responses.

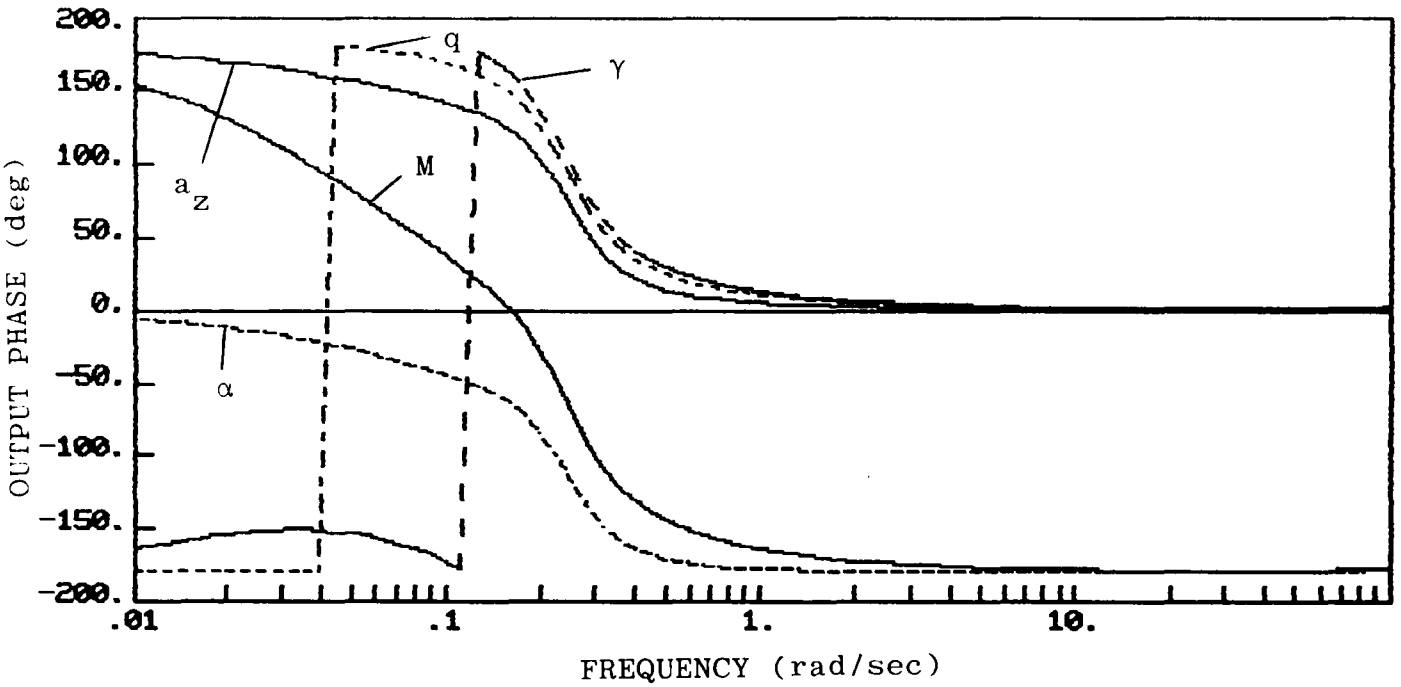

Figure 5-19. Three State Reduced Order Compensator Phase Responses.

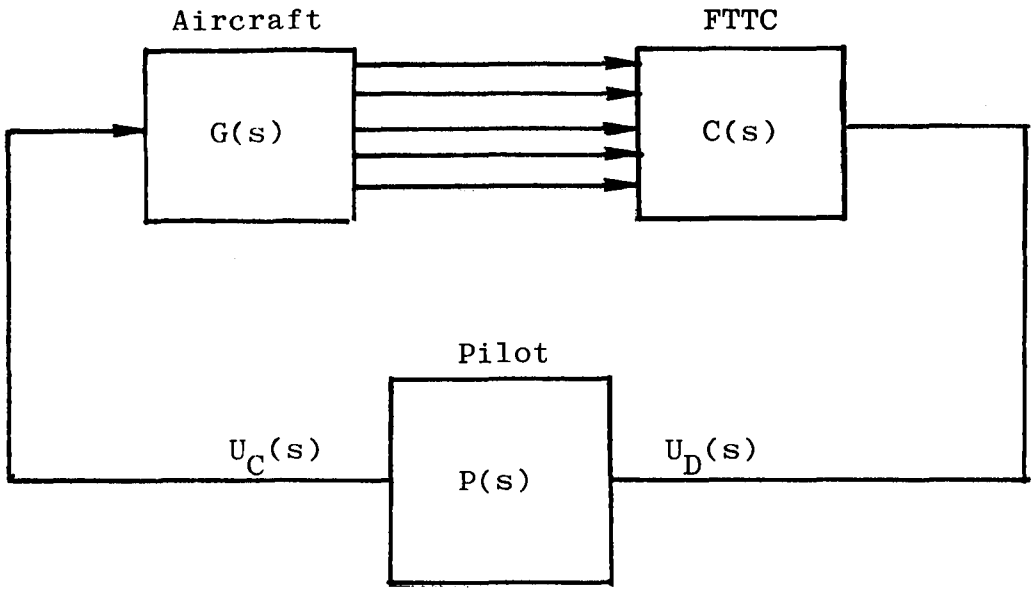

Figure 5-20. Controlled Plant Dynamics

The state equations for the controlled plant are

$$\begin{bmatrix} \dot{x}_c \\ \dot{x}_{Ac} \end{bmatrix} = \begin{bmatrix} F_c & , & G_c H_{z_{Ac}} \\ O & & F_{Ac} \end{bmatrix} \begin{bmatrix} x_c \\ x_{Ac} \end{bmatrix} + \begin{bmatrix} O \\ G_{Ac} \end{bmatrix} u \quad ,$$

$$U_D = H_c x_c \quad .$$

The frequency response of the controlled plant is shown in Figures 5-21 and 5-22. In the region of .5-2 radians/sec the plant is k_1/s^2 and in the region of 2.-20 radians/sec it is approximately k_2/s^3. A k/s^3 plant is clearly harder to control than a k/s^2 plant, however the plant gain at the upper range of these frequencies is so low (shown on the log-log scale here) that the change in dynamics is not significant. The pilot model is based on negative feedback, hence the phase of the $\frac{-1}{s^2}$ plant should ideally remain close to zero, and as shown in Figure 5-22 remains between -20° to -50° in the .5-2 radian/sec frequency range.

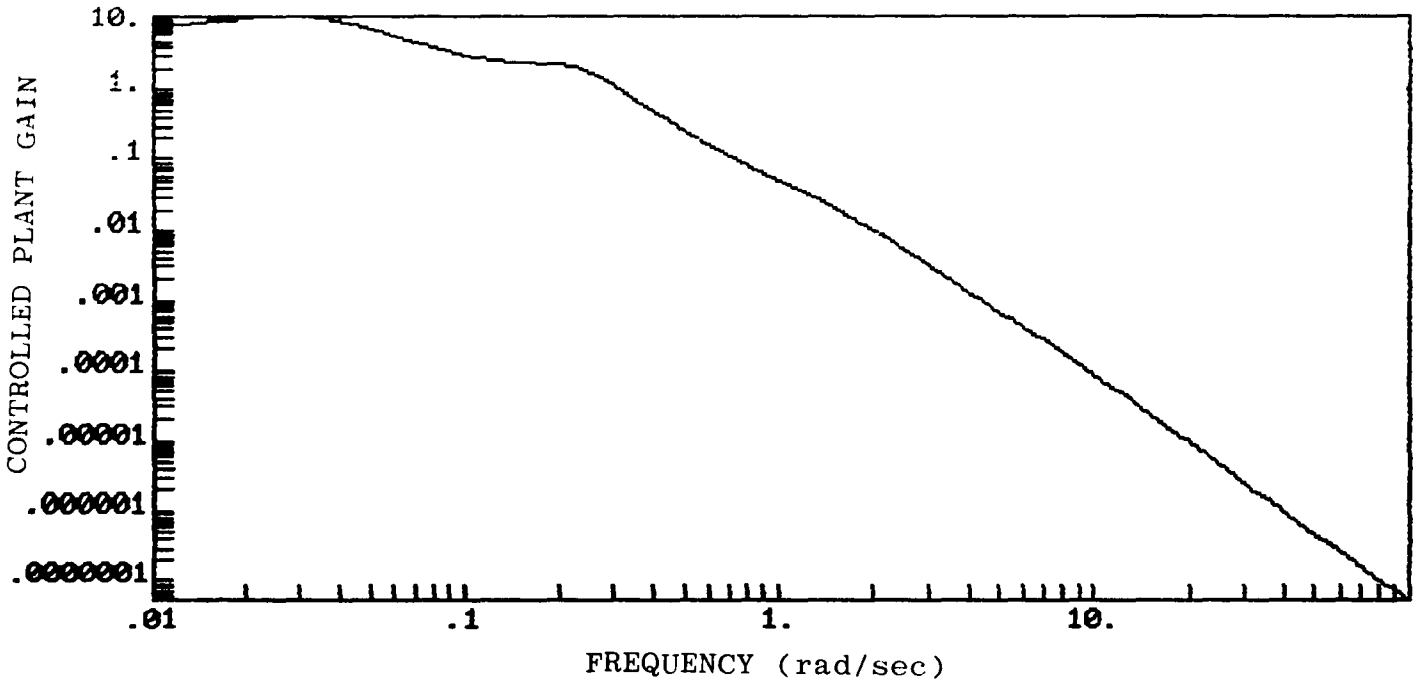

Figure 5-21. Controlled Plant Magnitude Response
(9 State A/C + 3 State Compensator)

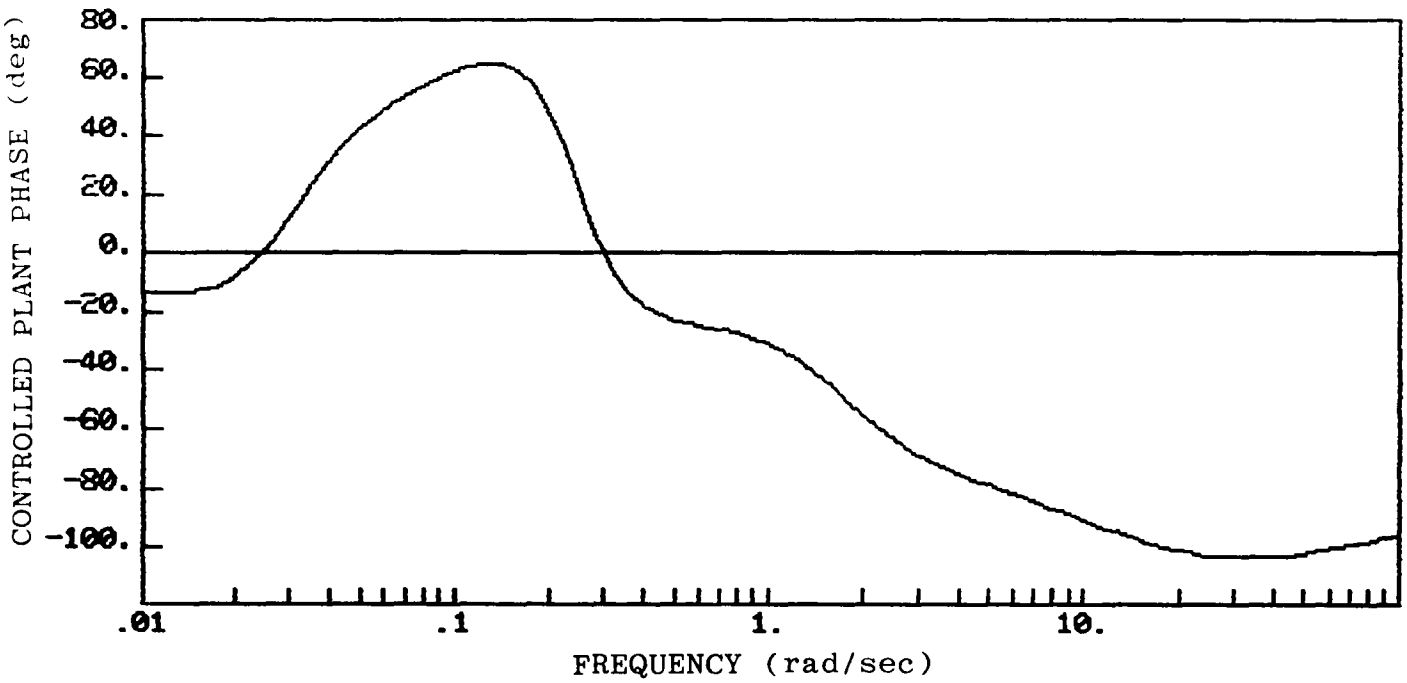

Figure 5-22. Controlled Plant Phase Response
(9 State A/C + 3 State Compensator)

5.4 LINEAR CONTROL LAW EVALUATION

Before evaluating the compensator in the nonlinear simulation the compensator must meet linear evaluation tests for stability, disturbance rejection, and other desirable controller properties. The effect of sampling rates can be evaluated much more inexpensively on the linear model than with extensive nonlinear simulation.

5.4.1 Stability

Stability evaluation of the compensator can be systematically checked by computing the eigenvalues of

1) the full-state compensator and design model (these should be the regulator and estimator eigenvalues)

2) the full-state compensator and evaluation model

3) the reduced order compensator and the design model, and

4) the reduced order compensator and the evaluation model.

In each case an augmented dynamics matrix is formed

$$\begin{bmatrix} \dot{x}_c \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} F_c & G_c H_z \\ GH_c & F \end{bmatrix} \begin{bmatrix} x_c \\ x \end{bmatrix} ,$$

and the eigenvalues computed. The eigenvalues for the second and final tests are shown in Table 5-7 below.

5.4.2 Disturbance Rejection

The performance of the regulator alone assuming full state feedback in rejecting gust disturbances can be evaluated with the following equations

$$(F-GC)X + X(F-GC)' = -\Gamma Q_{gust} \Gamma^T$$

$$Y = H_y X H_y .$$

22nd Order Evaluation Model	16th Order Evaluation Model
-19.8241 + 0.0000i	-19.8241 + 0.0000i
-14.0000 + 0.0000i	-14.1338 - 0.0000i
-5.0023 - 0.0000i	-9.8726 + 1.0553i
-1.2176 + 1.2470i	-9.8726 - 1.0553i
-1.2176 - 1.2470i	-7.6555 + 0.0000i
-1.0000 - 0.0033i	-5.0024 - 0.0000i
-1.0000 + 0.0033i	-1.2176 + 1.2470i
-0.0666 - 0.2403i	-1.2176 - 1.2470i
-0.0666 + 0.2403i	-0.0668 + 0.2410i
-0.3298 - 0.0000i	-0.0668 - 0.2410i
-0.1208 - 0.0000i	-0.3298 - 0.0000i
-0.0272 - 0.0271i	-0.1206 + 0.0000i
-0.0272 + 0.0271i	-0.0270 + 0.0271i
-0.0024 + 0.0000i	-0.0270 - 0.0271i
-14.0000 - 0.0000i	-0.0025 + 0.0000i
-10.0013 + 0.0000i	0.0000 + 0.0000i
-9.5222 - 0.0000i	
-8.0006 + 0.0000i	
-10.0000 + 0.0000i	
-9.5240 - 0.0000i	
-8.0000 - 0.0000i	
0.0000 + 0.0000i	

Table 5-7. Closed Loop Eigenvalues

The rms values are the square roots of the diagonals of Y, shown below with a gust spectral density of .0369 deg^2-sec:

```
YRMS =

  1.0737D-01   g
  1.3364D-01   deg
  4.8379D+00   deg/sec
  8.9651D-02   deg
  1.0981D+00   .001 Mach

URMS =

  6.5074D-02   inches (Longitudinal stick motion)
```

This control law has not been optimized but can be improved with additional design iterations.

5.4.3 Sampling Rate Effects

The effect of sampling rates can be evaluated by discretizing the aircraft model as well as the pilot model and FTTC compensator, at the least common multiple of all the sampling rates involved. The dynamics augmented with two delays for the display update and fifteen delays for the sensor update and the appropriate discrete feedback compensation, can then be evaluated for average behavior by solving the discrete form of the symmetric Lyapunov equation.

The discrete model for the compensator ($1/\Delta$ = 160 Hz) is,

$$x_c(k+1) = F_D x_c(k) + G_D z(k)$$

$$u(k) = H_D x_c(k) \ ,$$

where these matrices are given by

```
FD   =

   9.9879D-01   1.3969D-03   8.3799D-04
  -1.4818D-03   1.0013D+00   4.0402D-04
  -3.4794D-03   1.8787D-03   9.9841D-01

GD   =

  -5.5877D-03   9.7859D-03  -1.2077D-02  -4.0745D-02   5.0991D-03
  -7.4538D-03   1.4178D-02  -1.2112D-02  -3.3695D-02   2.6332D-03
  -3.6836D-03   5.0136D-03  -1.3074D-02  -5.3270D-02   8.6910D-03

HD   =

   4.4421D-01  -2.4000D-01  -1.8743D-01 .
```

The eigenvalues in the z-plane of the compensator are:

$$z_i = \begin{matrix} 9.9949D\text{-}01 + 1.4343D\text{-}03i \\ 9.9949D\text{-}01 - 1.4343D\text{-}03i \\ 9.7951D\text{-}01 - 3.8201D\text{-}21i \end{matrix}.$$

The discrete model for the pilot (1/Δ also 160 Hz) is

```
FD   =

   9.3941D-01  -2.1628D+00  -6.8223D-03  -1.3613D-05
   0.0000D+00   9.9996D-01   6.2375D-03   1.8290D-05
   0.0000D+00  -1.9510D-02   9.9408D-01   3.6609D-03
   0.0000D+00  -6.0384D+00  -1.8396D+00   8.1562D-01

GD   =

   5.4052D-03
   5.8515D-04
  -1.2520D-02
   2.2370D-01

HD   =

  -5.6103D+01   0.0000D+00   0.0000D+00   0.0000D+00
```

with z-plane eigenvalues,

$$z_i = \begin{matrix} 9.3941D\text{-}01 + 0.0000D+00i \\ 9.5123D\text{-}01 - 3.3148D\text{-}17i \\ 9.4221D\text{-}01 + 5.6503D\text{-}17i \\ 9.1622D\text{-}01 - 1.9440D\text{-}17i \end{matrix}$$

5.5 NONLINEAR SIMULATION CONTROL LAW EVALUATION

After linear evaluations, the control law can be more significantly evaluated by demonstration in the nonlinear simulation. This required closing the control loop of the simulation; the controls no longer need to be open-loop commands, but rather can be feedback from the simulation outputs, corrupted by a desired noise level. The details of how this FTTC simulator can be invoked are given in the next section.

The zoom and pushover control law developed in this section was implemented on the modified version of NASA Dryden's SIMII simulation. Following the algorithm development steps outlined in subsection 5.1:

Steps (1)-(2): The F-15 trim values were chosen for the desired apex conditions

α_o = 11 degrees

z_o = 32,000 feet

Steps (3)-(4): Assume an acceleration of V_o^2/R_o = 10 feet per second to climb a thousand feet in approximately 15 seconds. Thus, the radius of curvature is approximately 20,000 feet (for a V_o^2 that is 90% of the trimmed value).

The relations in Section 5.1 give V_o = 406 feet/second and an initial flight path angle of γ_i = 17.55°. This is a little too far away from the linearization of the aerodynamics at 11°. Fixing γ_i at 11 degrees gives the trajectory parameters shown in Table 5.8. The linear model trim condition is approximately half way in the maneuver velocity, and hence dynamic pressure range.

TABLE 5-8 ZOOM AND PUSHOVER DEMONSTRATION TRAJECTORY PARAMETERS

$z_o - z_i$	1000 feet
R_o	52,933. feet
ξ	$6.147_{10} - 3$
V_o	442. feet/sec
M_o	.4491
T_o	5544.6 lb_f
$(g - a_n)$	3.706 feet/second^2
$t_o - t_i$	23.23 seconds
γ_i	11.°
V_i	510.4 feet/second
M_i	.5175

To simplify the $\alpha_c(t)$ command we keep it constant while decreasing the flight path angle linearily in time under a stabilized engine power lever angle condition. The thrust itself is not stabilized because it is also a function of Mach, as the aerodynamics are. This maneuver is an acceleration initial condition response, shown in Figure 5-28 through 5-32.

The loss of velocity (see Figure 5-32) causes a higher angle-of-attack and acceleration seen in Figure 5-28 and 5-29. The velocity is lost because it is difficult to trim the aircraft in a condition where the zoom and pushover can begin. The ballooning effect, seen in the flight path angle response together with the variation of the thrust with velocity accounts for the loss of energy. It reaches a stable angle-of-attack condition. Initial trim would provide near perfect response.

Use of throttle would give considerably better control, and would also enable modeling the trajectory commands as a linear system so that all the control loops could be closed. Random walk states could also be appended to model the effects of changing dynamic pressure and thrust. MIMO servomechanism control could also be used since the response is influenced by the trajectory commands significantly.

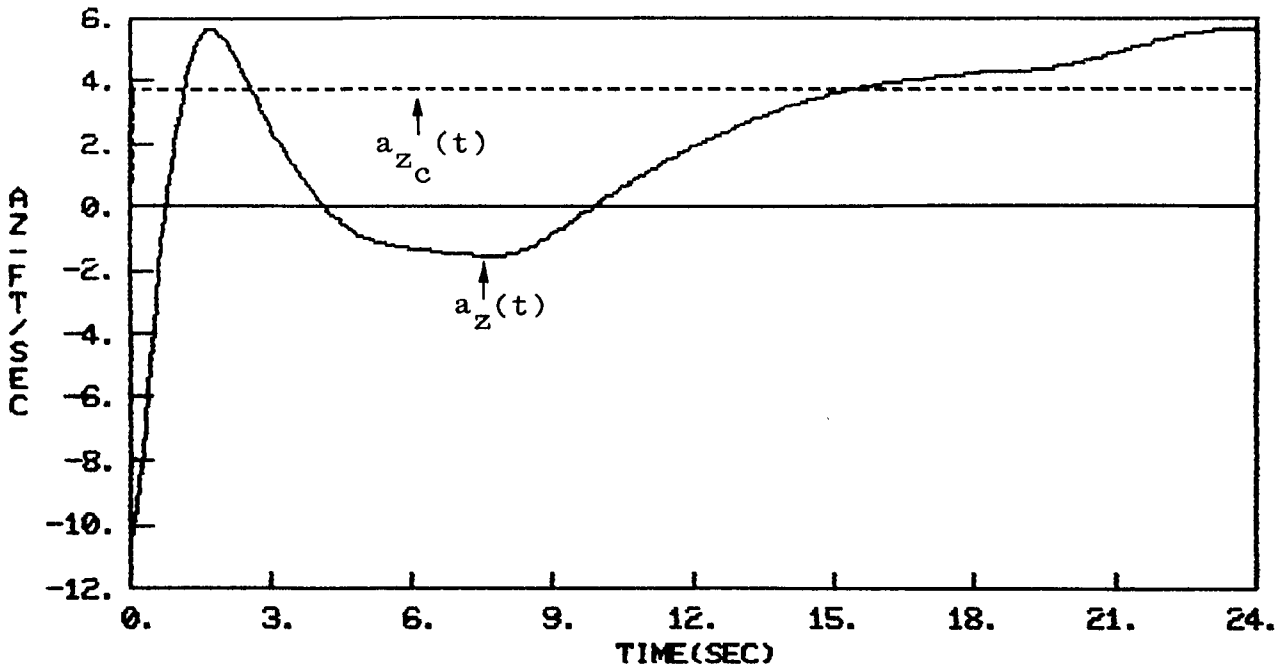

Figure 5-23 Vertical Acceleration - g's (a_z Initial Condition)

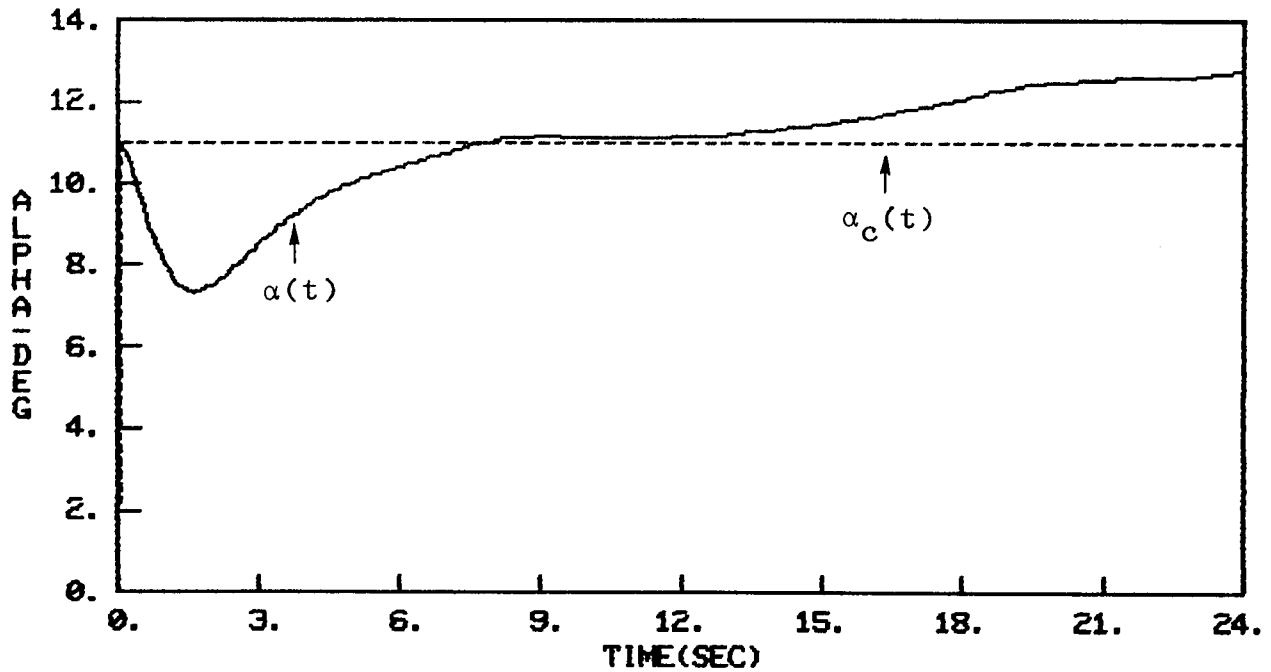

Figure 5-24 Angle-of-Attack - Degrees (a_z Initial Condition)

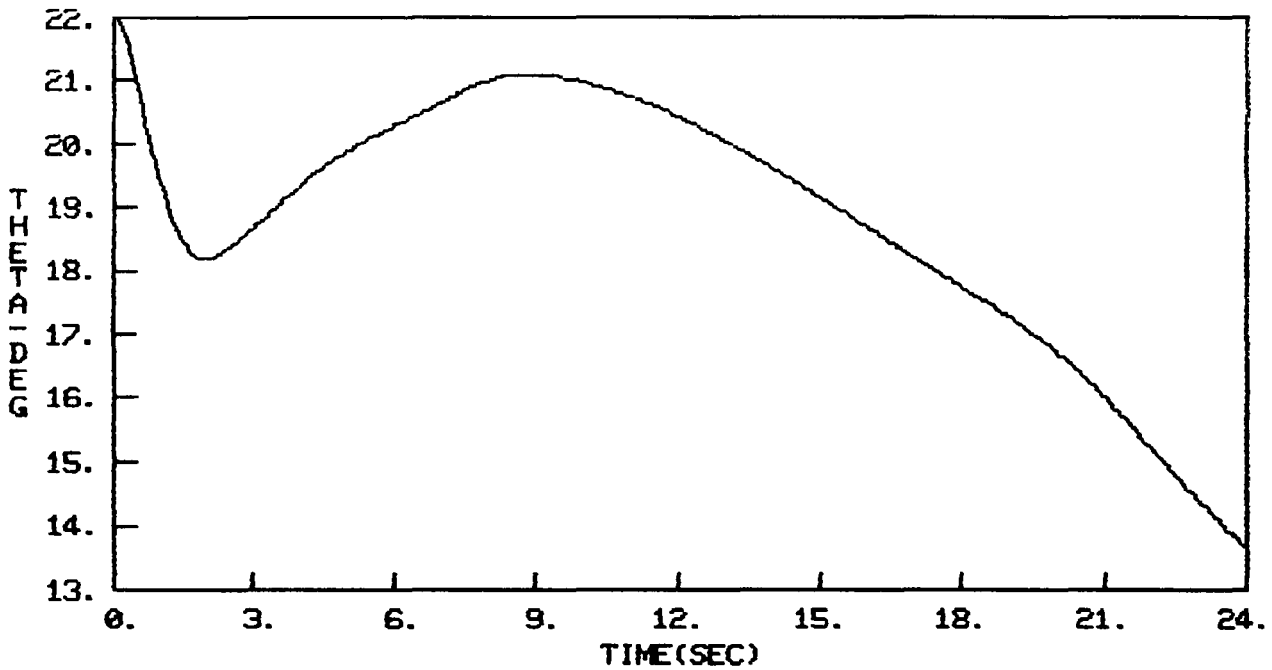

Figure 5-25 Pitch - Degrees (a_z Initial Condition)

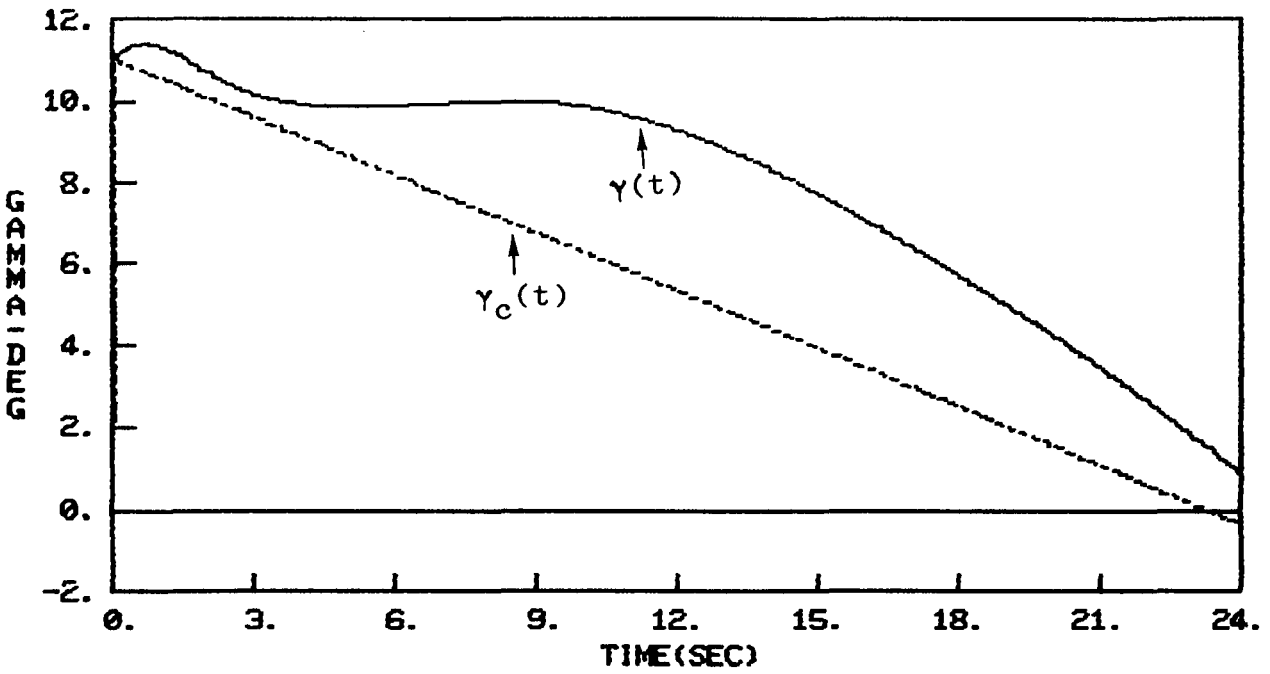

Figure 5-26 Flight Path Angle - Degrees (a_z Initial Condition)

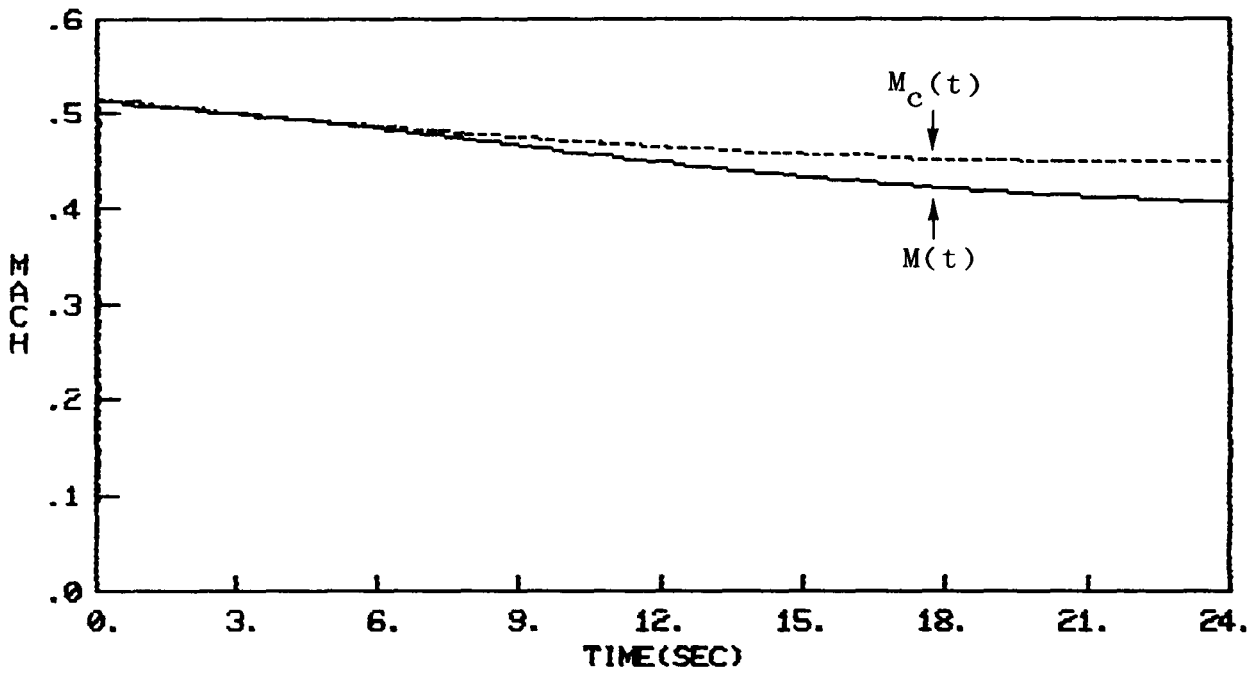

Figure 5-27 Mach (a_z Initial Condition)

5.6 ZOOM AND PUSHOVER ALGORITHM SUMMARY

The development of a performance index and trajectory commands with the necessary initial for the zoom and pushover maneuver was described in trajectory modeling section. Reduced order models for the aircraft and piloted aircraft were described and a linear compensator synthesized to regulate about the necessary trajectory commands. This compensator was reduced to three modes, and its stability and disturbance rejection properties evaluated with the full order linear model (thirteen states).

The compensator and pilot models were discretized and implemented in the nonlinear simulation as described in the next section and the FTTC User's Guide (Appendix C). The initial condition response illustrates the desirability of closing the loop on the trajectory commands as well. This demonstration used an open loop trajectory command generator, since the thrust was stabilized.

The particular control law, the linear compensator and the chosen trajectory command generators, is not optimized but illustrates all the aspects of the systematic design approach including the maneuver modeling, linear model development, linear control design, linear evaluation, and nonlinear evaluation with simulation.

SECTION 6

ZOOM AND PUSHOVER ALGORITHM DOCUMENTATION

The zoom and pushover maneuver FTTC is shown in block diagram form in Figure 6-1. The inputs and outputs used by each block are given along with references to sections of this report which document the equations implemented by each block.

For the zoom and pushover with a sensor data rate of 10 Hz and display update rate of 53-1/3 Hz, the simulation can be integrated at the least common multiple of the sampling rates, in this case 160 Hz. The discrete compensator equations are updated every sixteen aircraft integration steps and the display is updated every three steps. (Note that the multiple of integration time step for the sensors and displays can be input to accommodate general multirate systems.)

The compensator can have arbitrary structure including dynamics and feed forward gains. The pilot model is actually four independent models which affect the four pilot controls: elevator, aileron, rudder, and throttle. Any or all of these can be activated and can have identical or different structure and/or parameters which are defined with input data (see Appendix C).

Neglecting for a moment the dynamics of the compensator, the corresponding regulator control law is of the general form

$$u = - C_x(x - x_T - x_c) + u_T + u_c ,$$

where x_T are the trim states and x_c are the commanded states from the trajectory generator. The overall flight test trajectory controller is a linear feedback compensator, with set point control,

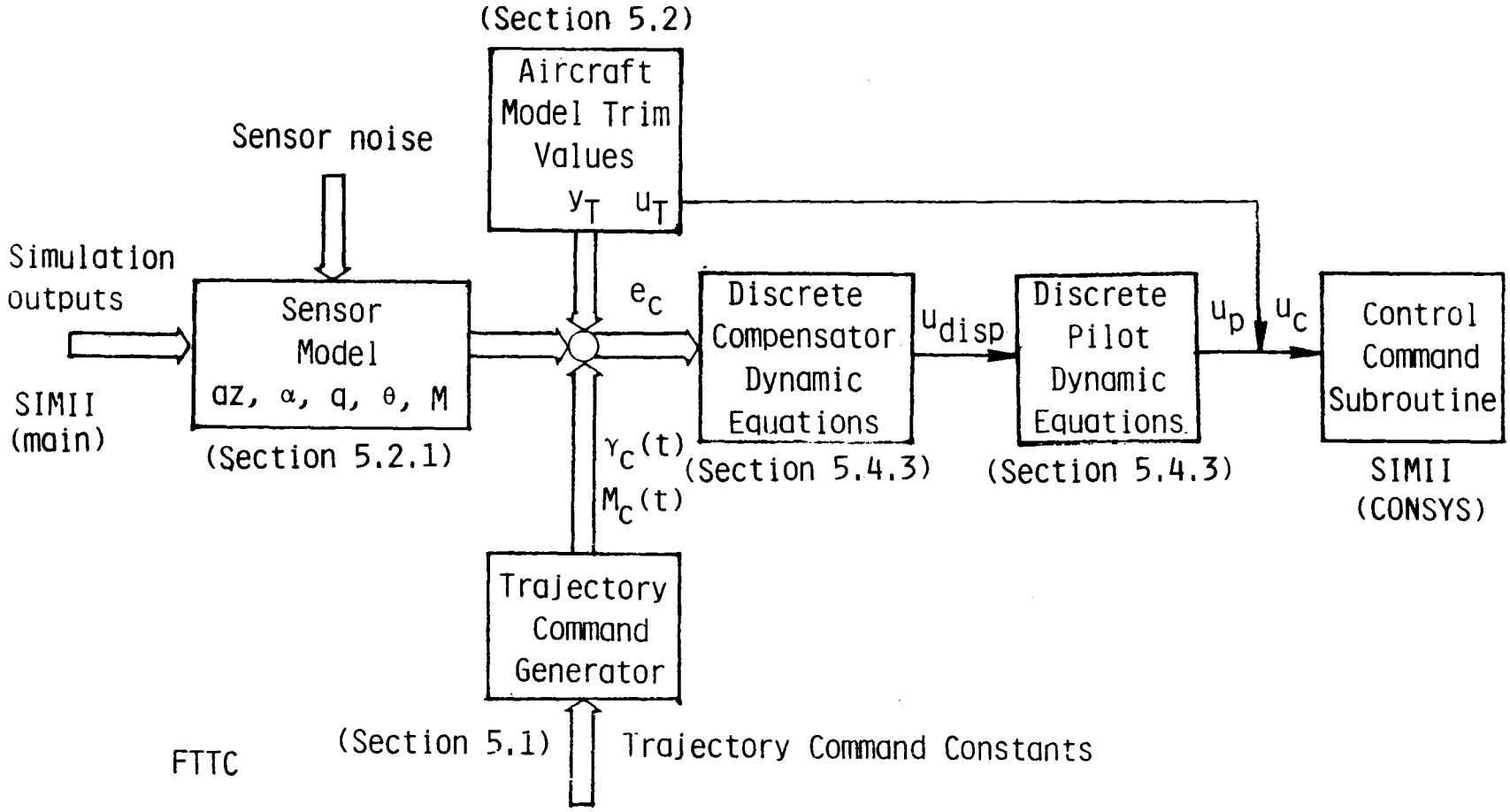

Figure 6-1 Zoom and Pushover Flight Test Trajectory Controller (FTTC) Block Diagram

and an open loop command generator. If we generated the trajectory commands as outputs of a linear system, these loops could be closed as well in a MIMO servomechanism. On the zoom and pushover maneuver, however, it is the constraint of stabilized thrust that requires the open loop trajectory commands.

This FTTC structure can be used for general simulation tasks as well. For example, the pilot model could be bypassed with pure feed forward unity gain, the sensors disconnected, and pure feed forward on the compensator to use the trajectory generator to input an open loop control function, such as a swept sine wave.

The flowchart for demonstrating the FTTC in the zoom and pushover maneuver, and any other maneuver, in a modified version of the SIMII simulation, is shown in Figure 6-2. The flowchart shows the sequence of operations in controlling a flight test maneuver and how the FTTC is interfaced with the NASA DRC simulation SIMII. The subroutine FTTC integrates the functions described in the block diagram (Figure 6-1). FTTC is called by the subroutine CNTRL when flight test trajectory simulation is requested, otherwise CNTRL implements an open loop control law, entered as segments of a time history. Only the subroutine associated with a particular sensor configuration (SENSOR) and a particular trajectory generator (TRAJGN) need to be recompiled for a different maneuver. All the others have a general structure suitable for all trajectories. These subroutine listings are given in Appendix D.

The outputs from the various blocks of the FTTC are shown in Table 1 of Appendix C. These user-accessible variables can be output from the simulation with other time history variables of interest.

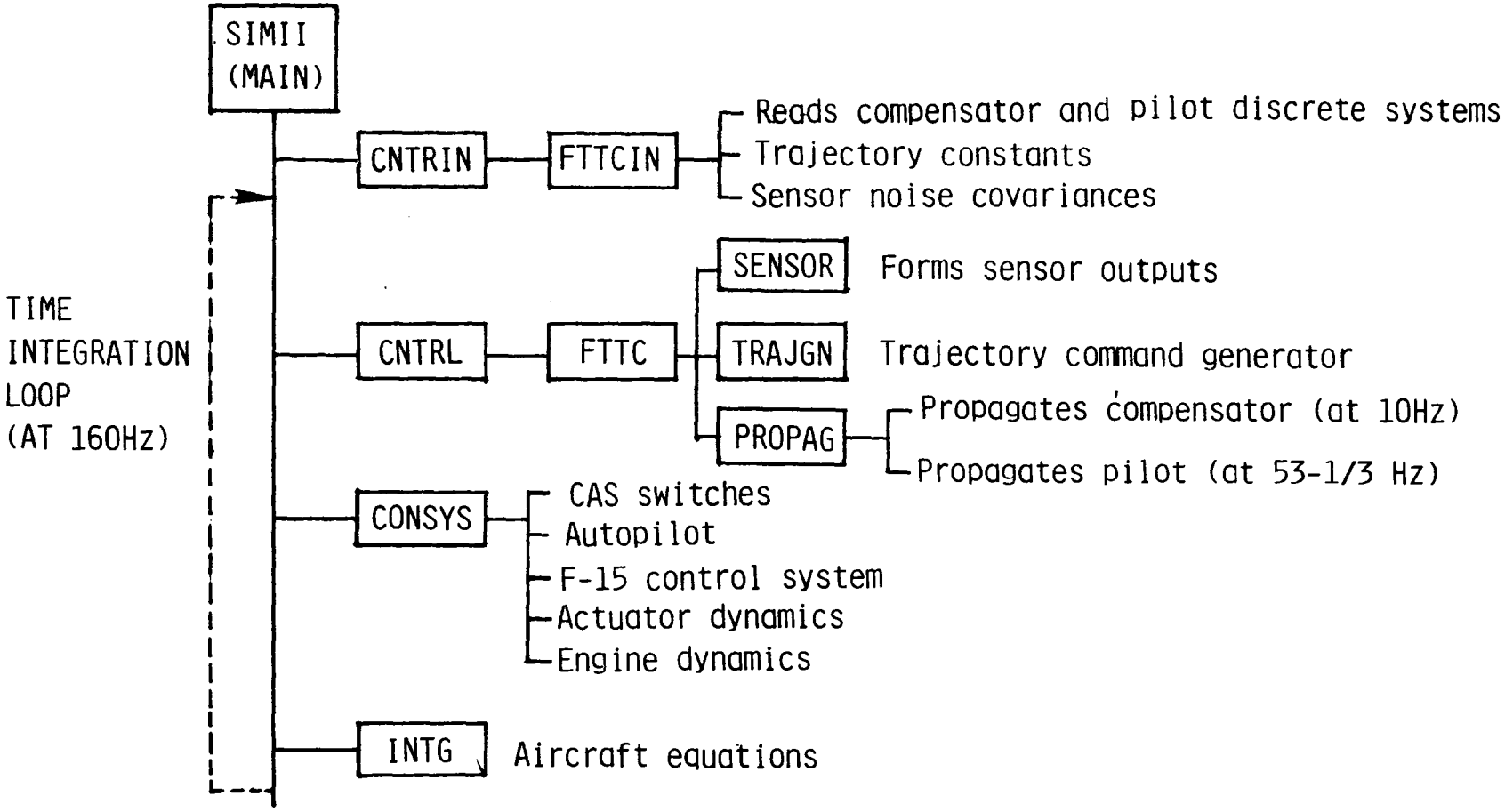

Figure 6-2. Flight Test Trajectory Controller Flowchart for the Batch Simulation (SIMII) Demonstration

SECTION 7

SUMMARY

Recent extensions to Linear-Quadratic-Gaussian (LQG) synthesis theory (Appendix A) applied to meaningful linear models (Section 5.2) with sufficiently flexible software tools (Appendix B) provide powerful techniques for designing Flight Test Trajectory Controllers (FTTC). The effort reported here has adapted and developed the necessary software and models to implement the zoom and pushover trajectory controller in the NASA-DRC batch simulation SIMII. The FTTC structure described in this report is sufficiently general to be used for other maneuvers, with the aid of the FTTC User's Guide [Appendix C]. The zoom and pushover maneuver controller design (Section 5) provides a detailed roadmap for the design procedure described in Section 4. This procedure involves a significant yet reasonable amount of engineering effort given sufficiently powerful software tools. Subsequent paragraphs summarize conclusions on the FTTC synthesis procedure, the linear models, the control design on the zoom and pushover maneuver, technical problems encountered and recommendations for further research.

7.1 SYSTEMATIC METHODS FOR FTTC DESIGN

The principal steps for systematic development of flight trajectory controllers can be summarized as planning, modeling, designing, and validating a trajectory controller. An important part of the planning is the trajectory specification. Major classes include single-vehicle and multiple-vehicle trajectories. Subclasses include constraints on combinations of load, speed

and altitude. General models suitable for trajectory control were demonstrated for the major subsystems, aerodynamics, kinematics, engines, sensors and pilot, by linearizing about the flight path. The control-design procedure consists of model reduction of the linear models where possible, design of a control law with a regulator/estimator or output feedback controller, and reduction of the controller complexity where possible for more robustness.

7.2 LINEAR MODEL DEVELOPMENT

Having access to a complete set of dimensional stability derivatives based on a detailed nonlinear F-15 aerodynamic model at arbitrary flight conditions is highly desirable. The specification of flight condition cannot be made completely arbitrary, however, because of the requirement to trim the aircraft, discussed further below. These dimensional stability derivatives are projected onto both stability and body axes. The longitudinal MCS/CAS, reduced to a fourth order model, is dependent on only one trim condition simulation parameter. To control high bank turn maneuvers the simulation has been extended to automatically trim in a steady coordinated turn. The significance of this maneuver is the difficulty for the pilot of maintaining altitude under high g-loads.

The simple linear pilot model represents the magnitude and phase response of a previously studied control-theoretic model of optimal human behavior. However, this is the most unreliable and variable part of the control loop. Suggestions for further research are given below.

7.3 CONTROL DESIGN FOR THE ZOOM AND PUSHOVER

Several conclusions can be made about design on a reduced order model. Significant feed forward terms result when there

a number of mid-frequency poles in which the designer is not strongly interested (e.g., the pilot model). Frequency shaped control methods are extremely helpful where the middle and high frequency model is obscured by model reduction. The feed forward term of the control into the sensor model is realized as a feedback loop in the compensator, and can have a significant effect on compensator dynamics.

Manipulation of models in a loop where each block is a linear system requires a flexible design program to form different block matrix equations which result from breaking the loop at different points. The abilities to quickly compute eigenvalues and frequency response plots are extremely helpful for catching modeling or equation errors, particularly once the physical interpretation of the model is obscured by model reduction.

7.4 TECHNICAL PROBLEMS

Since the validity of the linear models relies on a trimmed condition, the ability to get the SIMII simulation to trim near a desired flight condition can be a significant problem. Since the maneuvers of interest are all far from straight and level flight, a significant number of simulation runs were made to trim. A trimmed condition for the high bank turn necessary to obtain linear models for the level and windup turn maneuvers was not achieved during the contract. Hence, the linear control design methods have all been illustrated on the zoom and pushover maneuver. Changes to SIMII to help facilitate more efficient trim solution are described in the FTTC User's Guide (Appendix C). Further simulations, or identification not requiring trim, can provide these models.

The controlled plant transfer function is approximately $.05/s^2$. Yet increasing the pilot gain by 20. is too much to keep the low frequency dynamics from going unstable. Hence, additional work on pilot modeling and this gain adaptivity would be desirable.

7.5 RECOMMENDATIONS FOR FURTHER RESEARCH

Although robust controllers can be designed with existing pilot models, their lack of fidelity does not parallel detailed knowledge of the aircraft and its control system. This situation can in some sense be remedied by changing the structure of the FTTC to feed forward directly into the autopilot as well as through the pilot by driving a display. The task can be separated by frequency, giving the pilot his most favorable frequency band and letting the autopilot handle low and high frequency control. Such a controller can be developed by frequency weighting these two portions of the gain inversely with the general shape of the pilot model in a wide enough range that one has effectively, "safely" inverted a poorly known pilot in the loop to get higher performance with the controller.

Both the difficulty in achieving trim and all the engineering effort involved in modeling an MCS/CAS system can possibly be alleviated using linear identification and the known structure of the plant, identifying the aerodynamics involved in the maneuver and parameters of an assumed structure for the MCS/CAS.

It would be very desirable to validate an FTTC for a difficult maneuver, such as a high bank turn in a manned simulator after sufficient optimization of the controller in the batch simulation. Data could be collected in such a study for identification of the pilot model as well.

REFERENCES

[1] Kleinman, D., Baron, J. and Levison, W. H., "An Optimal Control Model of Human Response - Parts I, II," Automatica, 6, 1970, pp. 357-384.

[2] Shaked, U. and Karcanias, N., "The Use of Zeros and Zero-Directions in Model Reduction," Int. J. Contr., Vol. 23, 1976, pp. 113-135.

[3] Skelton, R. E., "Cost Decomposition of Linear Systems with Application to Model Reduction," Int. J. Contr., December 1980.

[4] Van Dooren, P., "A Generalized Eigenvalue Approach for Solving Riccati Equations," SIAM on Sci. and Stat. Comp., June 1981.

[5] Walker, R. A., Emami-Naeini, A. and Van Dooren, P., "A General Algorithm for Solving the Algebraic Riccati Equation," Proc. of 21st IEEE Conf. on Dec. and Control, Orlando, FL, Dec. 1982.

[6] Bryson, A. E. and Ho, Y. C., Applied Optimal Control, Hemisphere, Washington, D. C., 1975.

[7] Bryson, A. E. and Hall, W. E., "Modal Methods in Optimal Control Synthesis," Chapter in Vol. XVI of Advances in Dynamics and Control, Ed. by C. Leondes, 1980.

[8] Walker, R. A., "Computing the Jordan Form for Control of Dynamic Systems," SUDDAR 528, Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Stanford Univ., Mar. 1981.

[9] Gupta, N. K., "Frequency Shaping of Cost Functionals: Extensions of LQG Design Methods," AIAA J. of Guid. and Control, Vol. 3, No. 6, November-December 1980, pp. 529-535.

[10] Gupta, N. K., et al., "Modeling, Control and System Identification for Flexible Systems," AGARDograph on Spacecraft Guidance and Control, 1981.

APPENDIX A

FREQUENCY-SHAPING OF COST FUNCTIONALS

Consider state-variable model of the form

$$\dot{x} = Fx + Gu + w \ , \tag{A.1}$$

with state vector x, input vector u, and state-noise vector w. In steady-state LQG designs, a cost functional of the following form is minimized:

$$J = \lim_{T\to\infty} \frac{1}{T} \int_0^T (x^T Ax + u^T Bu)\, dt \ . \tag{A.2}$$

A is a positive-semidefinite matrix and B is a positive-definite matrix. In the frequency domain

$$J = \int_{-\infty}^{\infty} (x^* Ax + u^* Bu)\, d\omega \ , \tag{A.3}$$

where x in Eq. A.3 is the Fourier transform of time domain x. A and B are the same as in Eq. A.2. Note that Eq. A.3 specifies an equal penalty for states and inputs at all frequencies.

The problem with the time-domain approach is that it is not possible to specify frequency-dependent penalties. Therefore, in Ref. [9], the concept of frequency-shaping was developed in the frequency domain. We can make A and B functions of frequency in Eq. A.3. A technique for designing control laws for rational functional weightings has been developed [9]. The procedure requires appending additional states in LQG design problems. Instead of showing the mathematical details of the design approach, we will demonstrate it by an example.

Example: Consider a one-mode, undamped oscillator with state equations

$$\frac{d}{dt}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \ . \tag{A.4}$$

We shall select a frequency-shaped weighting function of the form

$$J = \int_{-\infty}^{\infty} (x^{*}ax + u^{*}bu)\, d \ , \tag{A.5}$$

$$a = \frac{1}{\omega^2 + 4} \ , \tag{A.6}$$

$$b = \omega^2 + 1 \ . \tag{A.7}$$

Defining

$$z = \frac{x}{j\omega + 2} \Rightarrow \dot{z} + 2z = x \ , \tag{A.8}$$

$$\bar{u} = (j\omega + 1)u = \dot{u} + u \ , \tag{A.9}$$

the cost functional becomes

$$J = \int_{-\infty}^{\infty} (z^{*}z + \bar{u}^{*}u)\, d\omega = \lim_{T\to\infty} \frac{1}{T} \int_{0}^{T} (z^2 + u^2)\, dt \ . \tag{A.10}$$

The appended state equations are a combination of Eqs. A.4, A.8 and A.9:

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ u \\ z \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -4 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & -2 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ u \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \bar{u} \ . \tag{A.11}$$

Standard LQG methods may be used to develop control laws for this modified problem. The designed controller will take the following form:

$$\bar{u} = C_1 x + C_2 \dot{x} + C_3 z + C_4 u \ . \qquad \text{(A.12)}$$

The implementation will use the equations obtained by substituting Eq. A.12 into the last two terms of Eq. A.11:

$$\frac{d}{dt}\begin{pmatrix} u \\ z \end{pmatrix} = \begin{pmatrix} -1+C_4 & C_3 \\ 0 & -2 \end{pmatrix}\begin{pmatrix} u \\ z \end{pmatrix} + \begin{pmatrix} C_1 & C_2 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x \\ \dot{x} \end{pmatrix} . \qquad \text{(A.13)}$$

Note that the controller is a low-pass filter. Thus, the region where the penalty on the state is small and the penalty on the control is high has reduced control activity. This behavior is typical of frequency-shaped control laws.

Notes:

1. The positive-semidefinite matrices A and B, provided they are functions of ω^2, may always be decomposed to convert a frequency-shaping problem into a modified LQG problem.

2. A standard Riccati equation needs to be solved to obtain the solution to the frequency-shaping problem.

3. The resulting controllers are usually dynamic and have memory. In fact, frequency-shaping is achieved through this memory.

4. The frequency shapes allowed by the techniques of Ref. [9] are a ratio of the polynomials in ω^2. A wide class of shapings may be approximated by such functions.

5. The frequency-shaping techniques shown above for the control design problem may also be extended for state estimation (see the following section).

A.1 ROBUSTNESS TO MODELING INACCURACY

Effects of inaccurate open-loop models on the closed-loop performance is managed by injecting minimum control power at natural frequencies of unmodeled modes. We will discuss procedures for controlling model inaccuracies in the high-frequency region, although similar techniques are applicable for other regimes.

The high-frequency robustness can be accomplished by modifying the state of the control weighting. The state weighting $A(j\omega)$ may be made a decreasing function of frequency. Three of the possible forms for $A(j\omega)$ are shown in the following:

$$(1) \quad A(j\omega) = \frac{1}{\omega^2} A \ , \tag{A.14}$$

$$(2) \quad A(j\omega) = \frac{\omega_o^4}{(\omega_o^2 + \omega^2)^2} A \ , \tag{A.15}$$

$$(3) \quad A(j\omega) = \frac{(\omega_1^2 + \omega^2)^2}{(\omega_o^2 + \omega^2)^2} A \ , \qquad \omega_1 > \omega_o \ . \tag{A.16}$$

To treat the frequency shaping of Eq. A.14, we define additional states $\bar{x}$ as follows:

$$\dot{\bar{x}} = x \ . \tag{A.17}$$

The performance index is

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T (\bar{x}^T \lambda \bar{x} + u^T B u) \, dt \ . \tag{A.18}$$

The control law will be of the form

$$u = C_1 x + C_2 \bar{x} \ . \tag{A.19}$$

The control gain C_2 will ensure that the high-frequency response is minimized. This formulation resembles integral control. However, the weighting matrices used in the design are different.

Spillover may also be reduced by placing high control weighting at high frequency. Examples of $B(j\omega)$ which reduce high-frequency spillover are as follows:

$$(1) \quad B(j\omega) = \frac{(\omega^2 + \omega_o^2)}{\omega_o^2} B \ , \tag{A.20}$$

$$(2) \quad B(j\omega) = \frac{\omega_1^2 + \omega^2}{\omega_o^2 + \omega^2} B \ , \qquad \omega_1 < \omega_o \ , \tag{A.21}$$

$$(3) \quad B(j\omega) = \frac{\omega^4}{[(\omega^2 + \omega_o^2)^2 + 2\omega_o^2 \omega^2]} B \ . \tag{A.22}$$

The frequency-shaped parts of these weightings are reciprocals of the frequency-shaped parts of state weightings. To implement the weighting function of Eq. A.20, we define a vector $\bar{u}$ as

$$\dot{u} + \omega_o u = \omega_o \bar{u} \ . \tag{A.23}$$

The performance index takes the form

$$J = \lim_{T \to \infty} \frac{1}{T} \int_0^T (x^T A x + \bar{u}^T B \bar{u}) \, dt \ , \tag{A.24}$$

The feedback control law will be

$$\dot{u} + \omega_o u = \omega_o \bar{u} = \omega_o (C_1 x + C_2 u) \ ,$$

or

$$\dot{u} + \omega_o (1 - C_2) u = \omega_o C_1 x \ . \tag{A.25}$$

This control law is shown schematically in Figure A.1.

During maneuvers and in other missions, there is often a need to not excite a particular mode at ω_o. Minimization of model errors at one frequency ω_o is achieved by:

$$A(j\omega) = (\omega^2 - \omega_o^2)^2 , \tag{A.26}$$

or by

$$B(j\omega) = \frac{B}{(\omega^2 - \omega_o^2)^2} . \tag{A.27}$$

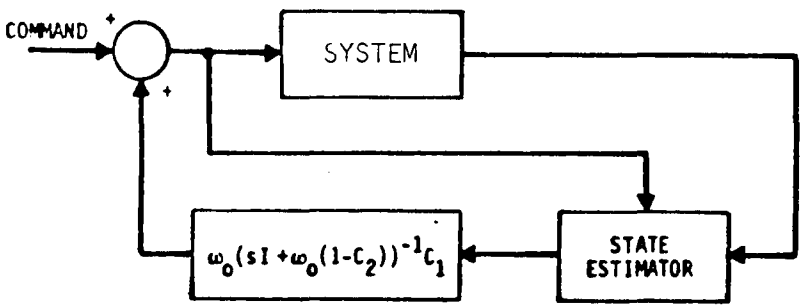

Figure A-1. A Frequency-Shaped Controller to Reduce High-Frequency Model Errors

A.2 ROBUST STATE ESTIMATION

The discussion will again address the estimation of low-frequency states when the high-frequency modes are not modeled. Extensions to other cases is straightforward.

In the linear systems of Eq. A.1, a linear transformation can be used to place all errors cause by unmodeled modes into the measurement equations. Note that when high-frequency modes are not modeled in the state equations, measurement errors in the modified system are also at high freqency. Therefore, the lumped modeling error at high frequency represents a high-frequency noise measurement with many peaks.

To derive a filter with desirable high-frequency behavior, we consider optimal estimation as the output of an optimization problem with the following performance index:

$$\bar{J} = \lim_{T\to\infty} \frac{1}{T} \int_0^T (w^T Q w + v^T R v)\, dt \ . \tag{A.28}$$

Again, conversion to frequency domain gives the following performance index:

$$\bar{J} = \int_{-\infty}^{\infty} (w^* Q(j\omega) w \ + v^* R(j\omega) v)\, d\omega \ . \tag{A.29}$$

The problem of model invalidity at high frequency is solved by making Q and R functions of frequency.

Since all errors associated with model truncation are incorporated in the measurements, we will consider frequency shaping only in R. High-frequency measurement spillover is controlled by increasing $R(j\omega)$ at high frequency. We may select $R(j\omega)$ as:

$$R(j\omega) = \frac{(\omega^2 + \omega_o^2)}{\omega_o^2} R \ . \tag{A.30}$$

We can define a new measurement $\bar{y}$ as

$$\dot{\bar{y}} + \omega_o \bar{y} = \omega_o y \ , \tag{A.31}$$

and

$$\bar{y} = H\bar{x} + v \ , \tag{A.32}$$

where

$$\dot{\bar{x}} + \omega_o \bar{x} = \omega_o x \ . \tag{A.33}$$

The state estimator is of the form:

$$\dot{x} = Fx + Gu = K(y - H\bar{x}) ,$$

$$\dot{\bar{x}} = -\omega_o\bar{x} + \omega_o x + \omega_o\bar{K}(\bar{y} - H\bar{x}) ,$$

$$\dot{\bar{y}} = -\omega_o\bar{y} + \omega_o y . \qquad (A.34)$$

This filter is shown schematically in Figure A-2. Note that a standard filter is obtained by setting $\bar{K}$ to zero and $\omega_o/(s + \omega_o)$ to one. In this formulation, filtered measurements are compared with filtered states. The eigenvalues of the closed-loop system are, in general, in the low-frequency region. Low-frequency, closed-loop eigenvalues reduce excitation of high-frequency modes through feedback in the state estimator. The filter shapes may be modified to provide any desired roll-off. The filter, of course, becomes more complex as faster roll-off is obtained. Note also that the transfer function between the estimated state and the measurements has at least two more poles than zeros. This filter has wide applicability in the design of practical state estimators.

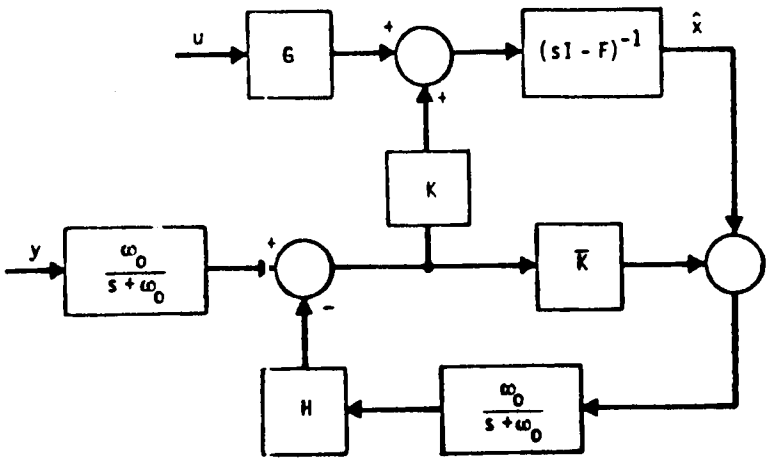

Figure A-2. A Frequency-Shaped Filter to Reduce High-Frequency Modeling Errors

A.3 DISTURBANCE REJECTION IN FLIGHT VEHICLES

Aircraft and rotorcraft are subject to high disturbance levels caused by engines or moving subsystems during in flight reconfiguration. In addition, turbulence, thrust variations, and density variations are external disturbance sources. These sources are nonwhite. Frequency-shaping methods may be applied to minimize the effect of these disturbances on the aircraft. Procedures for realizing disturbance rejection are shown in the following.

Consider a configuration transition of the F-111. The primary disturbance produced by the transition is at one frequency (typically the rotational frequency) and possibly its harmonics. Let the disturbance frequency be $\bar{\omega}$. Suppose we wish to minimize the effect of this disturbance on output y. This disturbance rejection can be realized by including the following term in the performance index:

$$\frac{1}{(\omega^2 - \bar{\omega}^2)^2}\, y^T A_y y \quad . \tag{A.35}$$

Note that the output penalty goes to infinity at the disturbance frequency. Thus, the impact of disturbance on the output is minimized.

The implementation of this frequency-shaped weighting requires definition of additional states as follows:

$$\dot{\bar{x}} = \begin{bmatrix} 0 & I \\ -\omega^2 I & 0 \end{bmatrix} \bar{x} + \begin{pmatrix} 0 \\ I \end{pmatrix} y \quad . \tag{A.36}$$

The performance index will take the form

$$J = \lim_{T} \frac{1}{T} \int_0^T (\bar{x}^T A_y \bar{x} + x^T A x + u^T B u)\, dt \quad , \tag{A.37}$$

and the control law is

$$u = C_1 x + C_2 \bar{x} \quad .$$

The flowchart for this control law is shown in Figure A-3.

The control formulation given above may be extended to include disturbance at other frequencies or over a frequency band. In each case, as shown in Figure A-3, there is memory in the control portion of the feedback.

A.4 SUMMARY

The application of frequency-shaping methods to flight vehicles leads to a linear controller with memory. However, the additional states needed to emulate frequency-dependent weights increase controller order. Software needed for these controller designs is similar to that for standard LQG problems.

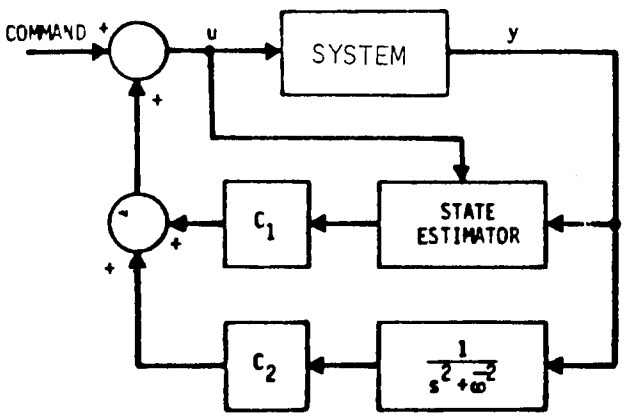

Figure A-3. A Frequency-Shaped Controller to Eliminate the Effect of Disturbance at $\bar{\omega}$ on Output y

APPENDIX B

$MATRIX_X$: A DATA ANALYSIS, SYSTEM IDENTIFICATION CONTROL DESIGN AND SIMULATION PROGRAM

Robert Walker, Charles Gregory, Jr., and Sunil Shah

$MATRIX_X$ is an interactive software system to perform a complete cycle of steps starting from data analysis to system identification, control design, simulation and evaluation. It is built on a user-friendly interpreter incorporating powerful matrix operations. State-of-the-art algorithms for linear system analysis, differential equation solutions, and Fourier transformation are included and can be called from the interpreter by simple commands. The package offers four major features

(1) Powerful interpreter, simple command structure, good graphics capability with most "book-keeping" chores handled by the software,

(2) State-of-the-art numerical algorithms, which allow solutions to high order problems,

(3) User transparent file management with uniform and consistent format, and

(4) Efficient implementation with a stack to require minimum memory and computation resources.

Basic algorithms for linear system solutions, eigensystem decomposition (including reliable determination of the Jordan form), singular value decomposition (SVD), QZ decomposition, and matrix algebraic operations are implemented as language primitives. Many of the primitives are inherited from $MATRIX_X$'s predecessor MATLAB, developed by Moler.

The interpreter can execute higher level constructs called macros and command files enabling construction of multi-level hierarchical structures. The command files are typically less than ten lines long and consist of interpreter primitives or other higher level constructs. Creation and modification of specific design and analysis procedures is easy because of the rich set of primitives and the hierarchical structure.

A set of command files and macros provide specific environments, such as data analysis and identification, control design and analysis, and simulation.

Data analysis and identification can be performed very efficiently and easily in $MATRIX_X$. Tied with a flexible graphics package, $MATRIX_X$ provides a production environment for batch and recursive identification methods. A universal interface with external simulations is provided to facilitate data transfer. Data can be censored, detrended and analyzed in $MATRIX_X$. Batch procedures include the standard regression methods with analysis of variance and step-wise regression. State-space and nonlinear batch maximum likelihood procedures are also available. Recursive algorithms such as the recursive maximum likelihood, extended Kalman filter with Ljung's modification, and recursive instrumental variable method are implemented. All covariance factorizations and updates are in U-D form for numerical reliability. Non-parametric batch and semi-batch methods using the FFT are provided for auto/cross covariances/spectras. Lattice updates for ARMAX systems are available. Adaptive control algorithms for multivariable systems using U-D updates can be designed using simple commands.

In $MATRIX_X$, control design can be based on any of the following:

(a) Linear Quadratic Gassian (LQG) approach,

(b) Methods based on A-B invariant subspaces,

(c) Eigenstructure assignment and zero placement, and

(d) Classical methods such as Nyquist and Bode.

For the LQG problem, the algebraic Riccati equation is solved from extended Hamilton equations avoiding inverses, which are troublesome in the singular case. The equations are row compressed with an orthogonal transformation followed by the QZ pencil decomposition and a backward stable ordering of the eigenvalues.

Meaningful extensions to LQG methods require inclusion of dynamics of reference, disturbances, sensors, and actuators. Appending of dynamics in frequency-shaped control design or model-following techniques involves forming augmented equations, which is easily accomplished with $MATRIX_X$ primitives. Use of frequency-shaped costs, with singular value plots for robustness evaluation, allow incorporation of engineering judgment in the control design.

Evaluation tools for linear systems include frequency response, power spectral density plots, time reponses, transmission zeroes and individual transfer function zeroes. The principal vector algorithm (PVA) primitive for numerically reliable extraction of the Jordan Form (with discriminatory rank deflation of root clusters) is very useful in modal analysis of open-loop systems of vehicles and structures. PVA permits computation of residues or partial fraction expansions of multivariable systems. Extraction of the Kronecker indices, supremal (A,B) invariant subspaces and the Kalman decomposition is performed using the SVD, QR and the QZ algorithm.

Simulation of stiff differential equations is performed with Gear's software. Linear system responses are computed efficiently from the residues.

Use of "chopped arithmetic," i.e., using various effective machine word-lengths can provide performance evaluation of on-board small word-length control system implementations.

Numerical reliability and stability are important in all of the analysis environments described above. Primitive matrix operations are based on the best available numerical software drawn from EISPACK, LINPACK, and recent research in numerical techniques.

Reporting and control of numerical errors is performed in several ways: Estimates of the problem condition number and the algorithms condition number are available for many of the primitives in $MATRIX_X$. When requested, $MATRIX_X$ provides an estimate of the problem condition, the algorithm condition and the solution accuracy when such an analysis has been presented in the current literature. If an error analysis has not been performed rigorously, condition estimates using varying word length arithmetic and perturbed initial data can be requested.

The program is coded in ANSI-77 FORTRAN and will run on any system allowing interactive execution of FORTRAN. The few machine dependent features are implemented separately for most popular operating systems. An implementation with 250K double-precision complex elements is available for VAX 11/780. System orders up to 150 can be handled easily within the storage capacity of the computer and numerical accuracy of the algorithms. Most algorithms access memory in a linear fashion, so that page-swapping is not unduly increased in the virtual memory implementation.

In summary, $MATRIX_X$ provides a system which minimizes engineering and programming resources required for the complete cycle of system identification, control design, and validation, yet provides the flexibility needed in an applied research environment.

APPENDIX C

FLIGHT TEST TRAJECTORY CONTROLLER USERS' GUIDE

AUTOTRIM CAPABILITIES

In the version of the NASA-Dryden Simulation SIMII modified for flight test trajectory control (FTTC), the autotrim function is quite important since the linear models depend on achieving trim. The previous four autotrim options are available with two new autotrims for steady coordinated turns. Level turns require a type of dynamic trim, since there are constant body angular rates, rather than zero angular rates. One of the turn trims specifies attitude, effective roll angle and angle of attack, adjusting the turn rate and velocity to achieve trim. The second turn trim is vice versa, specifying turn rate and velocity, adjusting the effective bank angle and angle of attack to achieve trim. A summary of constrained and adjusted variables for all six trim conditions is shown in Table 1 below.

Table 1 SIMII Trim Options

Trim Type	Program Mnemonic	Constrained Variables	Adjusted Variables
Normal	NORMAL, also 'ØØØØØØ'	V	PLA,α,β
Throttle	THRP	PLA	V,α,β
Angle-of-Attack	ALP	α	PLA,V,β
Sideslip	BTA	β	α,V,ϕ
Turnrate	PSIDOT	$\dot{\psi}_o$,V	PLA,α, ϕ_L(=>ψ,θ,ϕ)
Effective roll angle (angle of lift vector from the vertical)	PHILFT	α,ϕ_L(=>ψ,θ,ϕ)	PLA,V,$\dot{\psi}$

The gains which adjust the unconstrained variables at each autotrim iteration can be entered as input. Table 2 shows the appropriate user input variable and the associated default value for zero input.

Table 2 Autotrim Adjustment Gains and Input Parameters

Variable Location \ Trim Type/ Adjusted Variables	Normal	Throttle	α	β	$\dot{\psi}^\circ$	ϕ_L
UX(73)	PLA(.005)		PLA		PLA	PLA
UX(74)	α,β(.01)			α	α	$\dot{\psi}^\circ$
UX(75)		V(1.)		V		
UX(76)		α,β(.1)	β	ϕ(.1)	ϕ	
UX(77)			V(.3)			V
UX(79)	Starting value for PLA (Default = 0)					
UX(80)	Maximum number of iterations (10,000)					
UX(89)	$\dot{\psi}^\circ_o$ (Turn rate trim option) => p_o, q_o, r_o					
UX(90)	ϕ_L (Effective bank angle option), α_o => ψ_o, θ_o, ϕ_o					

The trim values can be preset to desired values without going through the autotrim iterations with the subroutine TRMSET. Values of the UX array are used as input. The association of variables and locations of UX are documented in this subroutine, included in Appendix D.

A summary of the user logical flags useful in FTTC design and evaluation are given below in Table 3.

Table 3 SIMII-FTTC Option Flags

Logical Input Flag	Option
LX(20) = T	Find Stability Derivatives (see Aerodynamic Linearization section)
LX(21) = T	Stop SIMII after finding stability derivatives
LX(22) = T	Bypass trim but set all trim values (invokes subroutine TRMSET)
LX(23) = T	Print out intermediate autotrim convergence data
LX(24) = T	FTTC closed loop simulation (additional inputs required (see Tables 4 and 5)
LX(25) - LX(28) = T	Invoke a pilot model for, respectively, the Aileron, Elevator, Rudder, and Thrust commands

AERODYNAMIC LINEARIZATION

When the simulation SIMII has been trimmed, all the stability derivatives can be requested as follows:

LX(20) = T Computes all the stability derivatives,

LX(21) = T Stops the simulation after the stability calculations

The six aerodynamic force and moment coefficients and the independent variables that affect them are shown in Table 5-2, the zoom and pushover trimmed example. Both stability axis and body axis variables are shown. The equations in SIMII are implemented with the force coefficients in the stability axes and moment coefficients in body axes. The linearization procedure perturbs the appropriate combinations of variables in stability axes to get the body axes derivatives. The effect of a center-of-gravity shift is included and the divided differences of the nondimensional coefficients are converted to dimensional stability derivatives before output.

The subroutine STABDR outputs the trimmed nondimensional coefficients along with the dimensionalizing coefficients (for checking total force and moment balances), sets up consistent velocity perturbations, and calls the routine SDCALC for each state variable of interest. SDCALC saves the initial coefficients in another six element vector, perturbs the passed state variable of interest, recalls the SIMII routine CCALC to find the perturbed six coefficients, finds the divided differences, appropriately dimensionalizes these stability derivative approximations, and prints out the state variable with its six stability derivatives.

FTTC INPUT AND OUTPUT

FTTC simulation is invoked by inputting LX(24) = T, with the necessary inputs summarized in Tables 4 through 6. Of the various blocks of the flight test trajectory controller (see Figure 6-1), the FTTCOM common block is used for linear models and the UX array is used to store the outputs of the various stages of the controller. Thus, time histories of the internal variables of the controller can be output for plotting.

Table 4 FTTC Input and Output Variables

Variable	Storage Location
Aircraft Model Trim Values	US(11) - US(25)
Sensor Noise Standard Deviations	US(26) - US(40)
Trajectory Constants	US(41) - US(55)
Compensator to SIMII Unit Conversions	US(56) - US(59)
Sensor Outputs	UX(11) - UX(25)
Trajectory Command Generator Outputs	UX(26) - UX(40)
Compensator Inputs	UX(41) - UX(55)
Compensator Outputs	UX(56) - UX(59)

The input for the compensator and pilot models is accomplished by FTTCIN, a subroutine called where the control input data is read, CNTRIN. Thus the FTTC input comes immediately after Group 11 data cards, the autotrim variables. The necessary FTTCIN cards are given in Table 5 below.

Table 5 FTTC Linear Model Input

Card	Column	Format	Description
1	1-5 6-10	2I5	Dimension of the compensator dynamics matrix (F_C), and the number of compensator inputs (i.e., NZ the number of sensors)
2	1-60	3D20.13	F_C, G_C, H_C, J_C, by rows (G_C by columns), a new card for each row (column). Note that there are <u>four</u> compensator outputs for the four principal controls
3	1-5	I5	Dimension of the first pilot model (e.g., Fp_A).
4	1-60	3D20.13	Fp_A, Gp_A, Hp_A, Jp_A by rows (G by columns), a new card for each row (column). Note that each pilot model is an independent SISO system
5 . . .	. . . .		Additional pilot models

Immediately following the FTTC linear system input, the necessary FTTC constants are input as shown in Table 6.

Table 6 FTTC Input Parameters

Card Type	Column Number	Format	Description	Variable Locations
1	1-5 6-10	2I5	Sensor and pilot model sampling intervals (i.e., multiple of simulation integration step)	NSAMPZ, NSAMPP
2	1-50	5E10.3	Aircraft trim values, y_T(NZx1) trim values plus four control trim values	US(11) - US(25)
3	1-50	5E10.3	Sensor noise standard deviations (assuming uncorrelated sensors)	US(26) - US(40)
4	1-5	I5	The number of trajectory parameters	NTRAJP
5	1-50	5E10.3	Trajectory parameter constants	US(41) - US(55)
6	1-50	5E10.3	Compensator to SIMII control unit conversions	US(56) - US(59)

APPENDIX D

FORTRAN LISTINGS FROM SIMII-FTTC

The new or significantly changed subroutines of the NASA DRC F-15 SIMII simulation are listed by function below. Their calling sequence and function are shown in Figure 6-2. The FORTRAN listings follow chronologically.

Autotrim Capabilities

AUTOTR	Adjust variables to achieve desired initial condition
TRMCHK	Check force and moment balances
TRMSET[1]	Set all trim values to desired variables

Aerodynamic Linearization

STABDR[1]	Perturb each state variable of interest
SDCALC[1]	Compute six stability derivatives for each state variable

FTTC Simulation

FTTC[1]	Integrate flight test trajectory control law
SENSOR[1]	Extract SIMII variables and add sensor noise
TRAJGN[1]	Trajectory command generator
PROPAG	Propagate discrete compensator and pilot models

$MATRIX_X$ Control Design Program Interface

MTOSIM	Translate $MATRIX_X$ data files with compensator and pilot linear models to SIMII-FTTC compatible file format
SIMTOM	Translate SIMII time history file to $MATRIX_X$ compatible file format for plotting

[1]Listing attached to this report

```
      SUBROUTINE TRMSET
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL  LX
C     ---- TRMSET SIMULATES A TRIMMED CONDITION BY SETTING THE VALUES
C          FROM UX(21)-UX(68)
C
      COMMON /VARDAT/ UX(90),LX(50)
      COMMON /DRVOUT/ F(13),DF(13)
      COMMON /SIMOUT/ AMCH,QBAR,GMA,DEL,UB,VB,WB,VEAS,VCAS
      COMMON /CONTRL/ DA,DAR,DAL,DFF,DH,DHR,DHL,DT,DR,DSB,THRST,THR,OMGE
      COMMON /CLCOUT/ CL,CM,CN,CD,CLFT,CY
      COMMON /CONPOS/ F1(9),DSBP,DFF,F2(4)
      COMMON /ENGDSP/ POSNZL,POSNZR,RPML  ,RPMR  ,TEMPL ,TEMPR ,
     .                FFLOWL,FFLOWR,FQNTY1
      LOGICAL FOUT,NOWC
      COMMON /ENGINF/ FQNTY,FFLOW,D,FOUT,NOWC,SINETA,COSETA
      DATA DTR/.0174532/
C
C     ---- UX 21-46 ARE F,DF
      DO 10 I=1,13
        F(I) = UX(20+I)
   10   DF(I) = UX(33+I)
C     ---- CONVERT TO INTERNAL UNITS ON ANGLE STATES
      DO 20 I=6,10
   20  F(I) = F(I)*DTR
C     ---- CONPOS VARIABLES UX 48-56,57-60
      DO 30 I=1,9
   30  F1(I) = UX(47+I)
      DO 40 I=1,4
   40  F2(I) = UX(56+I)
C     ---- CONTROL POSITIONS UX 61-64
      DA = UX(61)*DTR
      DH = UX(62)*DTR
      DT = UX(63)*DTR
      DR = UX(64)*DTR
C     ---- ENGINE PARAMETERS UX 65-68
      FFLOW = UX(65)
      THRST = UX(66)
      FFLOWL = UX(67)
      FFLOWR = UX(68)
C
      AMCH = UX(71)
      VEAS = UX(72)
      RETURN
      END
```

```
      SUBROUTINE STABDR
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL NORMAL,LONGIT,LATDIR,LEVEL
      COMMON /SIMOUT/ AMCH,QBAR,BMA,DEL,UB,VB,WB,VEAS,VCAS
      COMMON /ALTFUN/ A,RHO,G,PA
      COMMON /SIMTYP/ NORMAL,LONGIT,LATDIR,LEVEL
      COMMON /DRVOUT/ F(13),DF(13)
      COMMON /CLCOUT/ CL,CM,CN,CD,CLFT,CY
      COMMON /CONTRL/ DA,DAR,DAL,DFF,DH,DHR,DHL,DT,DR,DSB,THRST,THR,OMGE
      COMMON /SIMACC/ AX,AY,AZ,ANX,ANY,ANZ,AN
      COMMON /FMCOEF/ FM(39)
      COMMON /TRIGFN/ SINALP,COSALP,SINBTA,COSBTA,SINPHI,COSPHI,SINPSI,
     .                COSPSI,SINTHA,COSTHA
      COMMON /DATAIN/ S,B,CBAR,AMSS,AIX,AIY,AIZ,AIXZ,AIXE
      COMMON /STABCN/ QS,QSB
      EQUIVALENCE (T      , F( 1)),
     .            (P      , F( 2)),(Q     , F( 3)),(R     , F( 4)),
     .            (V      , F( 5)),(ALP   , F( 6)),(BTA   , F( 7)),
     .            (THA    , F( 8)),(PSI   , F( 9)),(PHI   , F(10)),
     .            (H      , F(11)),(X     , F(12)),(Y     , F(13))
C     ---------- STABILITY VARIABLE PERTURBATION VALUES
      DATA CVDEL,ALPDEL,BTADEL,PDEL,QDEL,RDEL,UDEL,VDEL,WDEL,ANZDEL,
     .     ADOTDL,DSBDEL,DADEL,DRDEL,DHDEL,DTDEL/16*.001/
      DATA CVNAM,ANAM,BNAM,PNAM,QNAM,RNAM,UNAM,VNAM,WNAM,AZNAM,ADNAM,
     .     DSBNAM,DANAM,DRNAM,DHNAM,DTNAM/'VELO','ALPA','BETA','P   ',
     .     'Q   ','R   ','U   ','V   ','W   ','ANZ ','ALFD','DSB ',
     .     'DA  ','DR  ','DH  ','DT  '/
C     ---------- FIND DIMENSIONAL CONSTANTS
      QS = QBAR*S
      QSB = QS*B
      WRITE(3,8900) AMSS,QBAR,QS,QSB
 8900 FORMAT(/,' AMSS,QBAR,QS,QSB = ',4(1X,1PE12.5),/)
      WRITE(3,8910) CL,CM,CN,CD,CLFT,CY
 8910 FORMAT(' NONDIMENSIONAL CL,CM,CN,CD,CLFT,CY = ',/6(1X,1PE12.5),/)
      WRITE(3,8920) (FM(I),I=1,39)
 8920 FORMAT(/,' CONSTITUENT COEFFICIENTS-SEE COMMON FMCOEF',/,
     . (6(1X,1PE12.5)))
C     ---------- SET UP VELOCITY PERTURBATIONS
      CVDEL = V*VDEL
      UDEL = CVDEL
      VDEL = CVDEL
      WDEL = CVDEL
      AMDEL = CVDEL/A
C     ---------- NOTE THAT STABILITY DERIV. ARE IN THE APPROPRIATE AXES
      WRITE(3,9000)
 9000 FORMAT('1DIMENSIONAL STABILITY DERIVATIVES FOR',/,
     .  9X,'CLBODY',7X,'CM',11X,'CNBODY',7X,'CD',11X,'CLFT',9X,'CYBODY')
C     ---------- FIND LONGITUDINAL DERIVATIVES
      IF(LATDIR) GO TO 100
C     ----ALPHA DER.
      CALL SDCALC(CL,ALPDEL,ALP,ALPDEL,V1,D1,.FALSE.,0,ANAM)
```

```
C     ---- Q DER.
      CALL SDCALC(CL,QDEL,Q,QDEL,V1,D1,.FALSE.,0,QNAM)
C     ---- W DER.
      CALL SDCALC(CL,WDEL,V,D1,ALP,D2,.TRUE.,2,WNAM)
C     ---- ANZ DER.
      CALL SDCALC(CL,ANZDEL,ANZ,ANZDEL,V2,D2,.FALSE.,0,AZNAM)
C     ---- ALPHADOT DER.
      CALL SDCALC(CL,ADOTDL,DF(6),ADOTDL,V2,D2,.FALSE.,0,ADNAM)
C     ---------- DERIVATIVES FOR BOTH LONGITUDINAL AND LATERAL DER.
  100 CONTINUE
C     ---- VELOCITY DER.
      CALL SDCALC(CL,CVDEL,V,CVDEL,AMCH,AMDEL,.TRUE.,0,CVNAM)
C     ---- U DER.
      CALL SDCALC(CL,UDEL,V,D1,ALP,D2,.TRUE.,1,UNAM)
C     ---- DH DER.
      CALL SDCALC(CL,DHDEL,DH,DHDEL,DSB,DHDEL,.TRUE.,0,DHNAM)
C     ---------- COMPUTE STRICTLY LATERAL DERIVATIVES
      IF(LONGIT) GO TO 200
C     ---- BETA DER.
      CALL SDCALC(CL,BTADEL,BTA,BTADEL,V2,D2,.FALSE.,0,BNAM)
C     ---- P DER.
      CALL SDCALC(CL,PDEL,P,PDEL,V2,D2,.FALSE.,0,PNAM)
C     ---- R DER.
      CALL SDCALC(CL,RDEL,R,RDEL,V2,D2,.FALSE.,0,RNAM)
C     ---- V DER.
      CALL SDCALC(CL,VDEL,BTA,D1,V2,D2,.FALSE.,3,VNAM)
C     ---- DA DER.
      CALL SDCALC(CL,DADEL,DA,DADEL,V2,D2,.FALSE.,0,DANAM)
C     ---- DR DER.
      CALL SDCALC(CL,DRDEL,DR,DRDEL,V2,D2,.FALSE.,0,DRNAM)
C     ---- DT DER.
      CALL SDCALC(CL,DTDEL,DT,DTDEL,V2,D2,.FALSE.,0,DTNAM)
  200 RETURN
      END
```

```
C ------------------------------------------------------------------------
      SUBROUTINE SDCALC(C,DELTA,VAR1,DELTA1,VAR2,DELTA2,MULVAR,IBODY
     .    ,NAME)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      LOGICAL MULVAR
      INTEGER NAME,IBODY
C     ---- MULVAR = F => ONLY ONE VARIABLE IS PERTURBED
C                   T => PERTURB TWO VARIABLES
C
C     ---- IBODY  = 0 => USE STAB. AXES FOR STAB. VAR. AND BODY AXES FOR BODY VAR.
C                   1 => PERTURB VT AND ALPHA FOR U
C                   2 => PERTURB VT AND ALPHA FOR W
C                   3 => PERTURB VT AND BETA FOR V
C
      DIMENSION C(6),CSAVE(6),DC(6)
      COMMON /CGSHFT/ DELX,DELY,DELZ
      COMMON /DRVOUT/ F(13),DF(13)
      COMMON /DATAIN/ S,B,CBAR,AMSS,AIX,AIY,AIZ,AIXZ,AIXE
      COMMON /STABCN/ QS,QSB
      COMMON /TRIGFN/ SINALP,COSALP,SINBTA,COSBTA,SINFHI,COSFHI,SINPSI,
     .                COSPSI,SINTHA,COSTHA
      EQUIVALENCE (V     ,F(5))
C     ----------  REINITALIZE AND SAVE COEFFICIENTS
      CALL CCALC
      DO 10 I=1,6
   10 CSAVE(I) = C(I)
C     ---------- PERTURB VARIABLES(ONE OR TWO)
      IF(IBODY .EQ. 0) GO TO 15
      GO TO(11,12,13),IBODY
C     ---------- PERTURBATION FOR U---NOTE THAT V IS TOTAL V(VT)
   11 DELTA1 = COSALP*DELTA
      DELTA2 = -SINALP*DELTA/(V*COSBTA)
      GO TO 15
   12 DELTA1 = SINALP*DELTA
      DELTA2 = COSALP*DELTA/(V*COSBTA)
      GO TO 15
   13 DELTA1 = DELTA/COSBTA
   15 VAR1 = VAR1 + DELTA1
      IF(.NOT.MULVAR) GO TO 20
      VAR2 = VAR2 + DELTA2
   20 CALL CCALC
C     ---------- FIND DIVIDED DIFFERENCES
      DO 30 I=1,6
   30   DC(I) = (C(I)-CSAVE(I))/DELTA
C     ---------- PREPARE TO ADD EFFECT OF C. G. SHIFT
      TERM = (DC(4)*COSBTA*SINALP + DC(6)*SINBTA*SINALP + DC(5)*COSALP)*
     .       QS
      TERM3 = QS*(DELX*(DC(4)*SINBTA - DC(6)*COSBTA) -
     . DELY*(DC(4)*COSBTA*COSALP + DC(6)*SINBTA*COSALP - DC(5)*SINALP))
C     ---------- MAKE DERIVATIVES DIMENSIONAL
      DC(1) = (DC(1)*QSB + TERM*DELY)/AIX
      DC(2) = (DC(2)*QS*CBAR - TERM*DELX)/AIY
```

```
      DC(3) = (DC(3)*QSB + TERM3)/AIZ
      DC(4) = -DC(4)*QS/AMSS
      DC(5) = -DC(5)*QS/AMSS
      DC(6) =  DC(6)*QS/AMSS
C     ---------- TRANSFORM CL AND CD TO BODY AXES
      IF(IBODY .EQ. 0) GO TO 40
      DCLB = COSALP*DC(5) - SINALP*DC(4)
      DCD  = SINALP*DC(5) + COSALP*DC(4)
      DC(5) = DCLB
      DC(4) = DCD
C     ---------- RESTORE THE INITIAL VALUES
   40 DO 50 I=1,6
   50   C(I) = CSAVE(I)
      VAR1 = VAR1 - DELTA1
      IF(MULVAR) VAR2 = VAR2 - DELTA2
C     ---------- PRINT OUT STABILITY DERIVATIVES
      WRITE(3,9000) NAME,DC
 9000 FORMAT(1X,A4,6(1X,E12.5))
      RETURN
      END
```

```
      SUBROUTINE FTTC(ITIME)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER ITIME
      COMMON /FTTCOM/FTT(200),IFC(3),IGC(3),IHC(3),IJC(3),
     .       IFPA(3),IFPE(3),IFPR(3),IFPT(3),IWK(3),
     .       NSAMPZ,NSAMPP
C
C     FTTC IMPLEMENTS A FLIGHT TEST TRAJECTORY CONTROL LAW AT
C     INTEGRATION INTERVAL ITIME
C
C     COMPENSATOR AND PILOT MODELS ARE STORED IN ARRAY FTT WITH
C     POINTER INDICES
C       I.E. IFC(1)    ROW DIMENSION OF FC (COMPENSATOR DYNAMICS)
C            IFC(2)    COL DIMENSION OF FC
C            IFC(3)    LOCATION IN FTT WHERE FC IS STORED
C             .
C             .
C             .
C            IFPT(1) ROW DIM. OF FPTH(PILOT DYNAMIC MODEL FOR THRUST)
C            IFPT(2) COL DIM. OF FPTH
C            IFPT(3) LOCATION IN FCC WHERE FPTH BEGINS
C       NOTE: FOR THE FIVE LINEAR MODELS THE CONVENTION IS TO STORE
C            F,G,H,J,X,Y (MODEL STATE AND OUTPUT) IN COMMON FTT
C
C       NSAMPZ  TIME INCREMENT FOR UPDATING MEASUREMENTS
C       NSAMPP  TIME INCREMENT FOR UPDATING PILOT MODEL
C
C       AIRCRAFT MODEL TRIM VALUES                 US(11)-US(25)
C       SENSOR NOISE STANDARD DEVIATIONS           US(26)-US(40)
C       TRAJECTORY CONSTANTS                       US(41)-US(55)
C       UNIT CONVERSIONS (COMPENS. TO SIMII)       US(56)-US(59)
C
C       SENSOR OUTPUTS                             UX(11)-UX(25)
C       TRAJECTORY COMMAND GENERATOR OUTPUTS       UX(26)-UX(40)
C       COMPENSATOR INPUTS                         UX(41)-UX(55)
C       COMPENSATOR OUTPUTS                        UX(56)-UX(59)
C
      COMMON /CONPOS/ DAP,FLAT,DATRIM,DEP,FLON,DETRIM,DRP,FPED,DRTRIM,
     .                DSBP,DFP,FLAPL,FLAPR,THSL,THSR
      COMMON /VARDAT/ UX(90),LX(50)
      COMMON /VARSIM/ US(90),LS(50)
C
C     ---------- INITIALIZE INPUT/OUTPUT HISTORY VARIABLES
      IF(ITIME .GT. 1) GO TO 10
      DO 5 I=11,59
    5   UX(I) = 0.D0
   10 CONTINUE
C
C     ---------- GENERATE TRAJECTORY COMMANDS
C
      IF(MOD(ITIME-1,NSAMPZ) .EQ. 0) CALL TRAJGN(US(41),UX(26))
C     ---------- UPDATE MEAS
C
      IF(MOD(ITIME-1,NSAMPZ) .EQ. 0) CALL SENSOR(UX(11),US(11),UX(26),
     .   UX(41),US(26),NZ)
C
C     ---------- PROPAGATE THE COMPENSATOR
```

```
C
   30 IX=IJC(3) + IJC(1)*IJC(2)
      IY=IX + IFC(1)
      CALL PROPAG(FTT(IFC(3)),IFC(1),FTT(IGC(3)),IGC(2),FTT(IHC(3)),
     .     IHC(1),FTT(IJC(3)),FTT(IX),UX(41),FTT(IY),FTT(IWK(3)))
C
C     ---------- SAMPLE AND HOLD THE COMPENSATOR OUTPUT
C
      IF(MOD(ITIME-1,NSAMPP) .NE. 0) GO TO 40
      DO 35 I=1,4
   35   UX(55+I) = FTT(IY-1+I)
C
C     ---------- PROPAGATE THE PILOT MODELS
C
   40 IF(.NOT. LX(25)) GO TO 50
C     ---------- AILERON PILOT MODEL
      IG = IFPA(3)+IFPA(1)*IFPA(2)
      IH = IG + IFPA(1)
      IX = IH + IFPA(1) + 1
      IY = IX + IFPA(1)
      CALL PROPAG(FTT(IFPA(3)),IFPA(1),FTT(IG),1,FTT(IH),1,FTT(IX-1),
     .     FTT(IX),UX(56),FTT(IY),FTT(IWK(3)))
C         ------ ADD A/C TRIM TO COMP. OUTPUT
      DAP = FTT(IY)*US(56) + US(11+NZ)
C
   50 IF(.NOT. LX(26)) GO TO 60
C     ---------- ELEVATOR PILOT MODEL
      IG = IFPE(3)+IFPE(1)*IFPE(2)
      IH = IG + IFPE(1)
      IX = IH + IFPE(1) + 1
      IY = IX + IFPE(1)
      CALL PROPAG(FTT(IFPE(3)),IFPE(1),FTT(IG),1,FTT(IH),1,FTT(IX-1),
     .     FTT(IX),UX(57),FTT(IY),FTT(IWK(3)))
C         ------ ADD A/C TRIM TO COMP. OUTPUT
      DEP = FTT(IY)*US(57) + US(12+NZ)
C
   60 IF(.NOT. LX(27)) GO TO 70
C     ---------- RUDDER PILOT MODEL
      IG = IFPR(3)+IFPR(1)*IFPR(2)
      IH = IG + IFPR(1)
      IX = IH + IFPR(1) + 1
      IY = IX + IFPE(1)
      CALL PROPAG(FTT(IFPR(3)),IFPR(1),FTT(IG),1,FTT(IH),1,FTT(IX-1),
     .     FTT(IX),UX(58),FTT(IY),FTT(IWK(3)))
C         ------ ADD A/C TRIM TO COMP. OUTPUT
      DRP = FTT(IY)*US(58) + US(13+NZ)
C
   70 IF(.NOT. LX(28)) GO TO 80
C     ---------- THRUST PILOT MODEL
      IG = IFPT(3)+IFPT(1)*IFPT(2)
      IH = IG + IFPT(1)
      IX = IH + IFPT(1) + 1
      IY = IX + IFPT(1)
      CALL PROPAG(FTT(IFPT(3)),IFPT(1),FTT(IG),1,FTT(IH),1,FTT(IX-1),
     .     FTT(IX),UX(59),FTT(IY),FTT(IWK(3)))
C         ------ ADD A/C TRIM TO COMP. OUTPUT
      THRP = FTT(IY)*US(59) + US(14+NZ)
```

```
C
   80 CONTINUE
C
      RETURN
C
      ENTRY FTTCIN
C
C     ---------- READ COMPENSATOR MATRICES
C
      READ(1,101) IFC(1),IGC(2)
  101 FORMAT(16I5)
      IFC(2) = IFC(1)
      IGC(1) = IFC(1)
      IFC(3) = 1
      IGC(3) = 1 + IFC(1)*IFC(2)
      IHC(1) = 4
      IHC(2) = IFC(1)
      IHC(3) = IGC(3) +IGC(1)*IGC(2)
      IJC(1) = 4
      IJC(2) = IGC(2)
      IJC(3) = IHC(3) + IHC(1)*IHC(2)
      NA = IFC(1)
      NZ = IGC(2)
      CALL READMA(FTT,NA,NA,NA,.TRUE.,'(3D20.13)   ')
      CALL READMA(FTT(IGC(3)),NA,NA,IGC(2),.FALSE.,'(3D20.13)   ')
      CALL READMA(FTT(IHC(3)),4,4,NA,.TRUE.,'(3D20.13)   ')
      CALL READMA(FTT(IJC(3)),4,4,IGC(2),.TRUE.,'(3D20.13)   ')
C
      WRITE(3,103)
  103 FORMAT(/,' ----- FTTC COMPENSATOR MODEL')
      CALL WRITMA(FTT,NA,NA,NA,'FC  ')
      CALL WRITMA(FTT(IGC(3)),NA,NA,IGC(2),'GC  ')
      CALL WRITMA(FTT(IHC(3)),4,4,NA,'HC  ')
      CALL WRITMA(FTT(IJC(3)),4,4,IGC(2),'JC  ')
C
      WRITE(3,8900) (LX(24+I),I=1,4)
 8900 FORMAT(/,' PILOT MODELS ARE ACTIVE FOR THE FOLLOWING CONTROLS-',/,
     .5X,'DAP   ',L1,/,5X,'DEP   ',L1,/,5X,'DRP   ',L1,/,
     .5X,'THRP  ',L1)
      IST = IJC(3)+IJC(1)*IJC(2)+IFC(1)+4
C
      IF(.NOT. LX(25)) GO TO 90
      WRITE(3,9010)
 9010 FORMAT(/,' ---- AILERON CONTROL PILOT MODEL MATRICES')
      READ(1,101) IFPA(1)
      NA = IFPA(1)
      IFPA(2) = NA
      IFPA(3) = IST
      CALL READMA(FTT(IST),NA,NA,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST),NA,NA,NA,'FPA ')
      N2 = NA*NA
      CALL READMA(FTT(IST+N2),NA,NA,1,.FALSE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2),NA,NA,1,'GPA ')
      CALL READMA(FTT(IST+N2+NA),1,1,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+NA),1,1,NA,'HPA ')
      CALL READMA(FTT(IST+N2+2*NA),1,1,1,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+2*NA),1,1,1,'JPA ')
```

```
      IST = IST+N2+3*NA+2
C
   90 IF(.NOT. LX(26)) GO TO 100
      WRITE(3,9020)
 9020 FORMAT(/,' ---- ELEVATOR CONTROL PILOT MODEL MATRICES')
      READ(1,101) NA
      IFPE(1) = NA
      IFPE(2) = NA
      IFPE(3) = IST
      CALL READMA(FTT(IST),NA,NA,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST),NA,NA,NA,'FPE ')
      N2 = NA*NA
      CALL READMA(FTT(IST+N2),NA,NA,1,.FALSE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2),NA,NA,1,'GPE ')
      CALL READMA(FTT(IST+N2+NA),1,1,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+NA),1,1,NA,'HPE ')
      CALL READMA(FTT(IST+N2+NA*2),1,1,1,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+2*NA),1,1,1,'JPE ')
      IST = IST +N2+3*NA+2
C
  100 IF(.NOT. LX(27)) GO TO 110
      WRITE(3,9030)
 9030 FORMAT(/,' ---- RUDDER CONTROL PILOT MODEL MATRICES')
      READ(1,101) NA
      IFPR(1) = NA
      IFPR(2) = NA
      IFPR(3) = IST
      CALL READMA(FTT(IST),NA,NA,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST),NA,NA,NA,'FPR ')
      N2 = NA*NA
      CALL READMA(FTT(IST+N2),NA,NA,1,.FALSE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2),NA,NA,1,'GPR ')
      CALL READMA(FTT(IST+N2+NA),1,1,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+NA),1,1,NA,'HPR ')
      CALL READMA(FTT(IST+N2+2*NA),1,1,1,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+2*NA),1,1,1,'JPR ')
      IST = IST + N2+3*NA+2
C
  110 IF(.NOT. LX(28)) GO TO 120
      WRITE(3,9040)
 9040 FORMAT(/,' ---- THRUST CONTROL PILOT MODEL MATRICES')
      READ(1,101) NA
      IFPT(1) = NA
      IFPT(2) = NA
      IFPT(3) = IST
      CALL READMA(FTT(IST),NA,NA,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST),NA,NA,NA,'FPT ')
      N2 = NA*NA
      CALL READMA(FTT(IST+N2),NA,NA,1,.FALSE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2),NA,NA,1,'GPT ')
      CALL READMA(FTT(IST+N2+NA),1,1,NA,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+NA),1,1,NA,'HPT ')
      CALL READMA(FTT(IST+N2+2*NA),1,1,1,.TRUE.,'(3D20.13)   ')
      CALL WRITMA(FTT(IST+N2+2*NA),1,1,1,'JPT ')
      IST = IST+N2+NA*3+2
C
  120 CONTINUE
```

```
      IWK(3) = IST
C
C     ---------- FTTC INPUT PARAMETERS
C
      READ(1,101) NSAMPZ,NSAMPP
      WRITE(3,108) NSAMPZ,NSAMPP
  108 FORMAT(/,' ---- SAMPLING MULTIPLES OF INTEGRATION TIME FOR THE',/
     . '      COMPENSATOR AND PILOT',10X,2I5)
      WRITE(3,109)
  109 FORMAT(/,' ---- AIRCRAFT TRIM VALUES')
      CALL READMA(US(11),1,1,NZ+4,.TRUE.,'(5E10.3)    ')
      CALL WRITMA(US(11),1,1,NZ,'YTRM')
      CALL WRITMA(US(11+NZ),1,1,4,'UTRM')
C
      WRITE(3,107)
  107 FORMAT(/,' ---- SENSOR NOISE STANDARD DEVIATIONS')
      CALL READMA(US(26),1,1,NZ,.TRUE.,'(5E10.3)    ')
      CALL WRITMA(US(26),1,1,NZ,'RZ  ')
C
      READ(1,101) NTRAJP
      WRITE(3,106)
  106 FORMAT(/,' ---- TRAJECTORY PARAMETERS')
      CALL READMA(US(41),1,1,NTRAJP,.TRUE.,'(5E10.3)    ')
      CALL WRITMA(US(41),1,1,NTRAJP,'TRAJ')
C
      WRITE(3,105)
  105 FORMAT(/,' ---- CONTROL UNIT CONVERSIONS (COMP. TO SIMII)')
      CALL READMA(US(56),1,1,4,.TRUE.,'(5E10.3)    ')
      CALL WRITMA(US(56),1,1,4,'UNIT')
C
C     ---------- INITIALIZE FOR SENSOR NOISE
      N = 0
      ST = GRAND(N)
C
      RETURN
      END
```

```
      SUBROUTINE SENSOR(Z,ZTRIM,ZC,EC,RZ,NZ)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Z(1),ZTRIM(1),ZC(1),EC(1),RZ(1)
      COMMON /DRVOUT/ F(13),DF(13)
      COMMON /SIMOUT/ AMCH,QBAR,GMA,DEL,UB,VB,WB,VEAS,VCAS
      COMMON /SIMACC/ AX,AY,AZ,ANX,ANY,ANZ,AN
C
C     Z(I) = UX(10+I)
C
C     ZTRIM(I) = US(10+I)
C     ZC(I)    = UX(25+I)
C     EC(I)    = UX(40+I)
C     RZ(I) = US(25+I)
C
C
C     ---------- OBTAIN SPECIFIC MEASUREMENTS HERE FROM SIMULATION
C                  OUTPUTS
C     ---------- START SPECIFIC ZAPO CODE HERE (SEE FTTC RPT SEC 5)----
C                  UNITS-- FT/SEC AND .01 RAD, RAD/SEC, ETC. FOR
C                          THE LINEAR ZAPO MODEL
C
C                Z(1) -- VERT. ACC.
C                Z(2) -- ANGLE OF ATTACK
C                Z(3) -- PITCH RATE
C                Z(4) -- PITCH ANGLE
C                Z(5) -- MACH
C
C
C     --------- FORM SENSOR MEASUREMENTS IN SIMII UNITS
C
      Z(1) = ANZ + COS(F(8))*COS(F(10))
      Z(2) = F(6)
      Z(3) = F(3)
      Z(4) = F(8)
      Z(5) = AMCH
C     ---------- FORM ERROR SIGNAL IN COMPENSATOR UNITS
      EC(1) = (Z(1)    -ZTRIM(1) - ZC(1))*32.174
      EC(2) = (Z(2)   -ZTRIM(2) - ZC(2))*100.
      EC(3) = (Z(3)   -ZTRIM(3) - ZC(3))*100.
C         ------ TURN GAMMAC INTO THETAC
      EC(4) = (Z(4)   -ZTRIM(4) - (ZC(4)+ZC(2)))*100.
      EC(5) = (Z(5)   -ZTRIM(5) - ZC(5))*1000.
C     ---------- END SPECIFIC ZAPO CODE HERE ----
C
C     ---------- ADD MEASUREMENT NOISE ( MUST HAVE COMPENSATOR UNITS!)
      DO 10 I=1,NZ
      ST = GRAND(N)
C     ---------- TEMP MOD TO SEE NOISE IN UX(61)-UX(65)
   10   Z(I) = Z(I) + ST*RZ(I)
      RETURN
      END
```

```
C ------------------------------------------------------------------
      SUBROUTINE TRAJGN(TRAJV,ZC)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION TRAJV(1),ZC(1)
C
C     TRAJGN GENERATES THE TRAJECTORY COMMANDS AND STORES THE
C       COMMANDED OUTPUTS IN ZC
C
      COMMON /DRVOUT/ F(13),DF(13)
      COMMON /SETICS/ FI(13)
C
C     ---------- THE FOLLOWING CODE MUST BE MODIFIED FOR DIFFERENT
C                  CLASSES OF MANEUVERS
C
C     ---------- START SPECIFIC ZOOM AND PUSH OVER CODE
C                (SEE FTTC FINAL REPORT SECTION 5.1)
C
C                TWO CONSTANTS AND THE DESIRED BIAS VALUES ARE THE
C                 7 TRAJECTORY VALUES
C                TRAJV(1) = AZO
C                     (2) = ALPHAO
C                     (3) = QO
C                     (4) = GAMMAO
C                     (5) = MO
C                     (6) = TFINAL
C                     (7) = G*RO/C**2
C
      ZC(1) = TRAJV(1)
      ZC(2) = DALPHA*F(1)/TRAJV(6) +TRAJV(2)
      ZC(3) = TRAJV(3)
      ZC(4) = -GAMAIC*F(1)/TRAJV(6) + TRAJV(4) + GAMAIC
      ZC(5) = DSQRT(TRAJV(7)*ZC(4)**2 + TRAJV(5)**2)
      RETURN
C
C     ---------- DO TRAJECTORY PARAMETER SETUP
C
      ENTRY TRAJST
      GAMAIC = FI(8)-FI(6)
      DALPHA = TRAJV(2) - FI(6)
C
C     ---------- STOP ZAPO SPECIFIC CODE
C
      RETURN
      END
```

1. Report No. NASA CR-170395	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FLIGHT TEST TRAJECTORY CONTROL ANALYSIS		5. Report Date February 1983 6. Performing Organization Code
7. Author(s) Robert Walker and Naren Gupta		8. Performing Organization Report No. ISI Report 16 10. Work Unit No.
9. Performing Organization Name and Address Integrated Systems Inc. 151 University Avenue, Suite 400 Palo Alto, California 94301		11. Contract or Grant No. NAS4-2906 13. Type of Report and Period Covered Contractor Report-Final
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code RTOP 505-31-44

15. Supplementary Notes

NASA Technical Monitor: Eugene L. Duke, Ames Research Center, Dryden Flight Research Facility

16. Abstract

Recent extensions to optimal control theory applied to meaningful linear models with sufficiently flexible software tools provide powerful techniques for designing flight test trajectory controllers (FTTCs). This report describes the principal steps for systematic development of flight trajectory controllers, which can be summarized as planning, modeling, designing, and validating a trajectory controller. The techniques have been kept as general as possible and should apply to a wide range of problems where quantities must be computed and displayed to a pilot to improve pilot effectiveness and to reduce workload and fatigue. To illustrate the approach, a detailed trajectory guidance law is developed and demonstrated for the F-15 aircraft flying the zoom-and-pushover maneuver.

17. Key Words (Suggested by Author(s))	18. Distribution Statement		
Trajectory guidance Pilot-in-the-loop control Maneuver modeling F-15 aircraft Display design Frequency-shaped optimal control	Unclassified-Unlimited STAR category 08		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 139	22. Price* A07

For sale by the National Technical Information Service, Springfield, Virginia 22161.

End of Document